A lightweight hybrid security framework for wireless sensor networks

Hichem Sedjelmaci, Sidi Mohammed Senouci

A Lightweight Hybrid Security Framework for Wireless Sensor Networks

Hichem Sedjelmaci and Sidi Mohammed Senouci
University of Bourgogne, DRIVE Lab
49 Rue Mademoiselle Bourgeois, 58000, Nevers, France
{Sid-Ahmed-Hichem.Sedjelmaci, Sidi-Mohammed.Senouci}@u-bourgogne.fr

Abstract— **On the one hand, intrusion detection systems (IDSs) have shown their ability to detect internal and external attacks with a high accuracy. On the other hand, cryptography techniques have proved their ability to assure the privacy of communication, i.e. data confidentiality. In this paper, we propose to develop and implement a lightweight Security Framework for Wireless Sensor Networks (WSNs) that combines the advantages of both cryptography and IDS techniques to assure the privacy of communication and detect the most dangerous attacks such as spoofed, altered or replayed routing information, man-in-the-middle, and denial-of-service attacks. However, the execution of both techniques at each node generates high overhead and energy consumption, even if we consider a cluster-based WSN. To economize nodes' energy, the cryptography operation is launched during a certain period determined mathematically and only within the cluster where a malicious node is detected. The proposed framework is evaluated analytically and implemented under TOSSIM simulator. According to our simulation results, our framework outperforms other security frameworks proposed in the literature in terms of detection rate, false positive rate, energy consumption and average efficiency.**

Keywords— Wireless sensor networks; Intrusion detection systems; Symmetric cryptography; Attacks

I. INTRODUCTION

Wireless sensor networks (WSNs) have attracted much attention due to their broad applications in military and civilian areas. They are an invaluable resource for realizing the vision of the Internet of Things (IoT). However, energy, memory constraints and the hostile environment in which they can be deployed make them more vulnerable to attacks. As a result, there is a strong need for security solutions that protect these kinds of networks from malicious attacks. In this context, intrusion detection systems (IDSs) have the capability to detect both internal and external attacks; unlike other security solutions such as cryptography, which simply prevents from external attacks to enter the network. Furthermore, cryptography techniques have the ability to assure data confidentiality, i.e. communication's privacy, unlike IDS techniques. Thereby, in this paper, we propose to develop and implement a new Lightweight Hybrid Security Framework in a cluster-based WSN (LHSFW), i.e based on the advantages of both intrusion detection and symmetric cryptography techniques.

The purpose of LHSFW is to prevent the occurrence of two kinds of attacks: (i) active attacks that aim to alter the packets' content of a target node or disrupting the smooth operation of a network. These kinds of threats are detected and removed from the network by applying a set of detection policies detailed in the next sections, and (ii) passive attacks, such threats that focus only on listening to the nodes' communication activity. In this case, LHSFW uses a symmetric cryptography technique to avoid the malicious node to eavesdrop the packets that pass within its radio range. However, running simultaneously these both techniques at each node generates a high overhead and degrades the network performances [1], even if we consider a cluster-based topology. Thereby, we propose a new security solution that aims to launch the cryptography operation only during a certain period determined mathematically and only within the cluster where a malicious node was detected. Furthermore, according to simulation results, we demonstrate that with the help of cryptography, LHSFW can remove almost all passive attacks from the network since such technique makes the information more attractive (i.e. by encrypting the message), which leads the passive attack to become an active one.

LHSFW aims to detect, using the IDS technique, several well-known and most dangerous active attacks such as spoofed, altered or replayed routing information, man-in-the-middle and denial-of-service attacks. Our approach outperforms other security frameworks proposed in the current literature [2][3], and exhibits a high accuracy of attacks detection with low energy consumption and high efficiency as demonstrated in simulation results.

The outline of this paper is organized as follows: In Section 2, we highlight some related works by indicating their advantages and weaknesses. In Section 3, we explain the characteristics of the most dangerous active attacks that occurred in WSN and detail the detection policy applied by the IDS to detect such attacks. Section 4 gives more details about the functioning of LHSFW framework and details its main components. Section 5 provides LHSFW's analytical and simulation results. Finally, conclusion and future work conclude this paper.

II. RELATED WORK

Recently, some researchers [2][3] aim to combine between the advantages of cryptographic technique defined as first line of defense and intrusion detection technique defined as second line of defense. The purpose of these hybrid mechanisms is to detect internal and external attacks with a high accuracy by using an intrusion detection technique, and assure the privacy

of communication by applying a cryptography technique. However, these mechanisms exhibit some weaknesses that are detailed in the following.

In [2], the authors conceive and implement an intrusion prevention and detection schema for cluster-based WSN. In the intrusion prevention protocol, a symmetric cryptography operation is used to assure an authentication of data source and provide a communication confidentially. In the intrusion detection protocol, the IDS agent monitors the nodes that are located within its radio range by applying a set of rules related to a set of attacks. According to experimental results, their schema exhibits a high detection rate when a certain number of attacks occur such as hello flooding, packet spoofing and impersonation attacks. However, both protocols work in a simultaneous way. Therefore, a high communication and computation overhead are generated, which leads to decrease the network lifetime.

In [3], the authors propose an energy efficient secure mechanism called eHIP, integrated in a LEACH (Low Energy Adaptive Clustering Hierarchy) protocol [4]. This mechanism applies an authentication algorithm to verify control message and sensed data. In addition, the authors propose a collaborative intrusion detection algorithm, where cluster members monitor their cluster-head and vice versa. According to simulation results, the authors claim that their approach is efficient in terms of energy conservation even when the number of sensors is important (i.e. scaling mode). However, this mechanism aims to secure the network against only one kind of denial of service attack.

In this paper, we propose to conceive and implement a new lightweight hybrid security mechanism that handles the weaknesses of the hybrid mechanisms proposed in the literature [2][3].

III. ATTACKS' DESCRIPTION AND DETECTION POLICY

Anomaly-based detection has the capability to detect unknown attacks by modeling only the normal behavior, and then identifying anything that deviates from this model as anomalous [5]. In this research, the IDS uses a specific category of anomaly detection defined as specification-based detection [6], which relies on a set of rules related to the most dangerous attack such as Spoofed, altered or replayed routing information, Man-in-the-middle and Denial of Service (DoS) attacks to model the normal behavior. The advantage of this detection policy is the ability to detect the attacks with a high accuracy and less energy consumption [6]. Therefore the LHSFW's intrusion detection module relies on this policy to detect the above attacks by using the rules cited above. We note that, these rules hold on the applications, where sensors continuously sensing the environment to send reports to the CH, which aggregates and forwards them to a base station.

A. Spoofed, altered or replayed routing information attack's detection rule

The intruder aims to monitor the communication, intercepts legitimate packets and thus alters or replays relevant information. The detection process of such attack is as follows: As illustrated in Fig. 1, during a time period, the IDS agent captures the same packets that are sent from A to B and B to CH, then computes the bytes' sum of each packet's payload, i.e. which are defined as: $Sp_{a\to b}$ and $Sp_{b\to ch}$. After that, the subtraction of these two results defined as sub is computed using equation 1. The IDS stores this value and carries out the same operation for each packet sent from A to B and B to CH. Finally, the normal distribution concept is used, by computing the standard deviations (SD) using equation 3. We note that, in a normal distribution concept, the normal node's behavior such as number of packets send or dropped, Received Signal Strength Intensity (RSSI) follow a normal distribution over time, i.e. the value of the behavior lies within (mean -3*SD) and (mean +3*SD)[7].

$$\text{sub}=Sp_{a\to b}-Sp_{b\to ch} \quad (1)$$

$$\text{mean (sub)}=\sum_{i}^{n}\frac{sub_i}{n} \quad (2)$$

In equation 2, we note that i ={1, ….., n} where n is the number of a same packet sent from A to B and B to CH.

$$SD(sub)=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(sub_i-mean(sub))^2} \quad (3)$$

The IDS attempts to determine whether these sums follow a normal distribution, as stated above by checking if the sub_i lies within (mean - 3*SD) and (mean + 3*SD). In contrast case, the node B is designed as intruder that carries out spoofed, altered or replayed routing information attack.

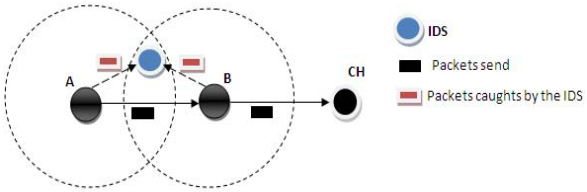

Fig. 1. Monitoring process of IDS agent

B. Man-in-the-middle attack's detection rule.

The malicious node makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection [8]. The attacker collects the packets from these victims' nodes and injects new ones. In addition, according to the wok undertaken by the authors in [9], they claim that such attack generates a high Received Signal Strength Intensity (RSSI) compared to it neighbors node. To detect such attack, first of all the IDS applies the previous detection rule (see subsection III.A) to check whether the same packet which is sent from A to B and B to CH is different or not by applying the normal distribution concept. When sub_i doesn't lie within the interval (mean - 3*SD) and (mean + 3*SD), the IDS compares RSSI values of the nodes located within its radio range; and the node with a higher RSSI is designed as an intruder that could carry out a man in the middle attack.

C. Denial of Service (DoS) attack's detection rule.

Such attack aims to exhaust the network resources or disrupt its proper operation. In the literature, a node that carries out a DoS attack can be launched by a variety of threats such as [10]: Selective forwarding, which removes an

important number of received packets, and Resource exhaustion that sends a considerable number of packets to waste the energy resource of targeted nodes. In this study, we focus to detect these two kinds of DoS: Selective forwarding and Resource exhaustion. The detection process of such attacks is as follows: the IDS agent monitors the number of sent and received packets by the nodes within a target link as illustrated in Fig. 1. Afterward, it computes the rate of sent packets over received packets of the nodes located within its radio range. In case, when a node (within a target link) exhibits a very high or very low rate compared to its neighbors, it will be designed as an intruder that could perform resource exhaustion or selective forwarding attacks, respectively.

IV. LIGHWEIGHT HYBRID SECURITY FRAMEWORK FOR WSN (LHSFW)

The proposed security framework, named LHSFW, aims to assure data's confidentiality and prevents the occurrence of the most dangerous attacks within a cluster-based wireless sensor network by using both symmetric cryptography and intrusion detection techniques. To improve the network lifetime, we focus to address the following issue: the cryptography and intrusion detection mechanisms don't run at all time in a simultaneous way within the same node as they incur a high communication and computation overhead [1]. Therefore, in our solution the cryptography operation is activated within a specific period (*Td*) only by the cluster members located within the same cluster as the malicious node detected by IDS agents.

We embed LHSFW in a cluster-based topology since it permits to increase the network lifetime compared to flat one. The purpose of a clustering protocol is to group a sensor into a set of cluster and elect at each one of them a node that has a high residual energy called cluster-head (CH). This later aggregate the received packets from it cluster members and forward the aggregated packet to the base station. Furthermore, the CH must coordinate the sensor nodes to ensure that the cluster reduces energy on average [11]. In this research, we use a Hybrid Energy Efficient Distributed protocol (HEED) [12] for the cluster formation and CH election. In order to simplify the process of intrusion detection and network's management, we fix the number of hops at each cluster equal to max two hops.

LHSFW is composed of two main components: the first and second security layers that apply respectively intrusion detection and symmetric key encryption techniques. These components are detailed in the following.

A. First security layer (FSL) component: Intrusion detection process

This component relies on an intrusion detection module, which is detailed in the following.

The IDS agent follows a detection policy, detailed in section III, to detect the attacker and prevent the occurrence of any malicious behavior within a cluster. When a suspected node is detected by the IDS, it sends a *Check* message to its corresponding CH to take a final decision about the behavior of this suspected node. This message includes the IDS'*id* and suspected node's i*d.* The CH computes a Malicious Behavior Level (MBL) of each suspected node detected by the IDSs and compares it with a trust formula (see equation 4). The MBL is computed as the number of IDS' detection over the number of IDS neighbors of the suspected node and it value belongs to the interval [0, 1]. When the MBL's value is more than a threshold (0.5) as shown in equation (4), this suspected node is designed as a malicious one. Therefore, the CH removes this attacker from the cluster and informs all the cluster members, i.e. by sending an *Encryption_message* to the second security layer (SL) component to activate and update their messages encryption operation and keys, respectively.

$$\begin{cases} \text{Node is Malicious if } 0.5 \leq \text{MBL} \leq 1 \\ \text{Node is Normal if } 0 < MBL < 0.5 \end{cases} \quad (4)$$

When a CH is suspected to be an attacker, the IDS agent sends a *Remove_broadcast* to the IDSs located within a neighborhood of this CH. This message includes the CH's id, Time To Live (TTL) and a malicious counter, incremented by 1 when CH exhibits a malicious behavior. We note that, TTL is defined as the number of IDS per each cluster over 2. When an IDS receives such message, it decrements by 1 the TTL value and checks the malicious behavior of the CH. When this attractive node is detected as an intruder, a malicious counter is incremented by 1, after that the IDS agent re-broadcasts such message to the other agents. This process will be completed when the two following situations occur: (i) malicious counter across a threshold *Th*, defined as the number of CH's IDS neighbors over 2. In this case an *Encryption_message* is forwarded to the SL component to launch the encryption operation and assure a data confidentiality (see SL component). Therefore, the CH is removed from the cluster and process of new CH's election is launched (see protocol HEED [12] for CH election). (ii) When the malicious counter doesn't reach the threshold *Th* and TTL value is equal to 0, such packet is removed. As a result, this process avoids the routing loops' issue (i.e. infinite packet circulation).

B. Second security layer (SL) component: symmetric key encryption

This component is integrated at each node to perform the encryption and decryption operations. In this research work, we propose a key distribution mechanism inspired from [13]. The process of the proposed key distribution is described as follows: Before, the nodes' deployment, we store a unique key $k\,primary$ at each node in order to crypt the keys sent by the nodes as described in the following. When the clusters are formed, the base station sends to each CH a key k_j where *j*= {1,…, k} and k is the number of clusters. After that, each CH generates keys $kchi_j$ from k_j for each CH's child using a hash function ($kchi_j$= hash ($k0_j$+*i*)) and sends for each one of them a key $kchi_j$ as illustrated in Fig. 2. We note that, *i*={1,…, c} and c is the number of CH's childs. At the end, the nodes that receive such key, compute a final key ki from this previous one (ki = hash ($kchi_j$)) and send it to the leaf nodes (i.e. nodes with zero child) as illustrated in Fig. 2. When the keys distribution is completed, the leaf nodes encrypt the message with *ki* and send this encrypted message to it corresponding CH's child, which decrypt it with this key, then send an

encrypted message (by using a key $kchi_j$) to its corresponding CH_j. Finally, the cluster-head decrypts the message and sends an encrypted one with a key k_j to the base station.

This component has the responsibility to assure the confidentiality of data, which is activated when a malicious node is detected. As explained above, in order to improve the network lifetime, the encryption and decryption processes are activated only within a cluster where an attack was detected by an IDS node. This component is equipped with two main modules: Key updating and data encryption modules.

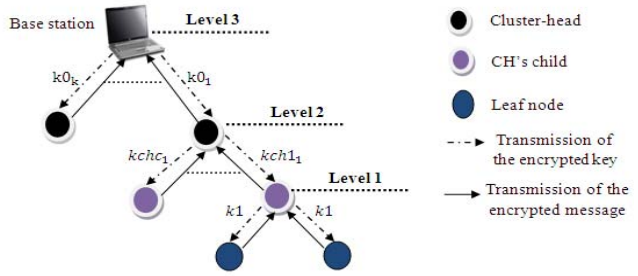

Fig. 2.Symetric key encryption: keys distribution and encryption process

1. Data encryption module: When CH confirms the malicious behavior of suspected node or malicious CH is detected by the IDSs (see intrusion detection module), the selected CH (a new CH in case when the older is an attacker) informs all nodes located within the same cluster, i.e. by sending an *Encryption_message,* to activate data encryption processes with computing new keys (see key updating module). We note that, this module uses a simple and fast cipher algorithm called Skipjack [14] to encrypt and decrypt data. As a result, an external passive attacker cannot eavesdrop the packets that pass thought its radio range, which allow data privacy within the network. It is important to note that, choosing Skipjack lies in the fact that it is very simple in both its operation and its key schedule and also it is the best algorithm in terms of memory requirements, with approximately 2,600 bytes of ROM [15].

In order to improve the energy consumption, when IDS agents located within the same cluster don't detect any malicious node during a time period *Td*; cluster members switch their exchanged messages from an encrypted one to a plaintext message. We note that the value of *Td* is determined mathematically in Subsection V.B.

2. Key updating module: This module has the responsibility to update the keys within a cluster where an attacker is detected. When a malicious node is detected within a cluster, the selected CH computes new keys $kchi\_s_j$ from the older ones for each CH's child, which are equal to $hash\ (kchi_j + n)$, where n is the number of nodes within the cluster. After that, it sends an *Encrypt_broadcast* message to each CH's child, which contains a key $kchi\_s_j$ encrypted by using a key $k\ primary\_s$=hash $(k\ primary + s)$. At the end, the CH's child computes a key ki_s, which is equal to hash $(kchi\_s_j)$ and sends this encrypted key to its leafs nodes (by using a key $k\ primary\_s$). We note that, s is the number of attacks detection at a cluster j by the IDSs. When this process is done, all the exchanged messages are encrypted during a time period *Td*.

V. PERFORMANCE EVALUATION

In this Section we embed and evaluate our framework under TOSSIM [16] and POWERTOSIM [17] simulators, a simulators of Tinyos sensor nodes. First, we determine the value *Td* defined as a required time to encrypt messages by nodes located within the same cluster before to switch to a plaintext message. In this study, we fix the number of passive attacks equal to 30 % of overall nodes. Afterward, LHSFW is compared to the two hybrid security mechanisms proposed in [2][3]. Specifically we compute the following metrics: detection rate, false positive rate, average efficiency and energy consumption, when the following active attacks occur: spoofed, altered or replayed routing information, man in the middle and denial of service attacks.

- **Detection rate (DR):** the ratio between the number of correctly identified attacks over the total number of attacks;
- **False positive rate (FPR)**: the ratio between the number of normal nodes that are incorrectly classified as malicious and the total number of normal nodes;
- **Average efficiency:** the required time for IDS agents to detect all attacks in the network, which is computed as follow: $AE = \frac{\sum_{i=1}^{n} E_i}{n}$ (5)

  where, E_i is the required time for the IDS agent to detect the occurrence of an attacker [5] and n is the number of attackers.

- **Energy consumption (EC):** the total amount of energy consumed by all sensors over the total number of nodes in the network.

A. Simulation setup

The main key parameters of the simulation are illustrated in Table I .We note that, in order to provide an accurate result, we carry out a set of 40 simulations, then compute the average of these simulation results (i.e. detection rate, false positive, energy consumption and average efficiency).

TABLE I. SIMULATION PARAMETERS

Simulation time	600 Seconds
Simulation area	100*100 m^2
Number of sensors	220
Number of clusters	10
Routing protocol	HEED
Radio range	15m
Sensor initial energy	5J
Passive attacks	30 % of overall nodes
Active attacks	From 5 % to 50% of overall nodes

B.Mathematic analysis of time's period Td

In this subsection, we first define the average number of malicious link's neighbors. Based on this, we determine the most appropriate value of time's period *Td* defined as the required time to encrypt messages by nodes located within the same cluster before to switch to a plaintext message.

The number of neighbors for the malicious link (i.e. nodes A and B) lie within the communication range of A and B as illustrated in Fig. 3. The area where the neighbors are located is calculated as follows [18]:

$$\text{area (x)}=2r^2cos^{-1}\left(\frac{x}{2r}\right)-\ 2x\sqrt{r^2-\frac{x^2}{4}}$$

The average number of neighbors for each malicious link is equal to E[area(x)]d and computed as follows:

$$E[\text{area(x)}]=\int_0^r \text{area(x)g(x)dx} \qquad (6)$$

The probability distribution of x is computed by

$$\mathrm{G(x)}=\frac{\mathrm{x}^2}{\mathrm{r}^2}$$

$\mathrm{g(x)}=\frac{\mathrm{dG(x)}}{\mathrm{x}}=\frac{2\mathrm{x}}{\mathrm{r}^2}$, which is the probability density function

$$E[\text{area(x)}]=\int_0^r\left(2r^2cos^{-1}\left(\frac{x}{2r}\right)-\ 2x\sqrt{r^2-\frac{x^2}{4}}\right)\frac{2\mathrm{x}}{\mathrm{r}^2}\ \mathrm{dx}$$

$$\Leftrightarrow E[\text{area(x)}]\approx 0.29\pi r^2 \qquad (7)$$

At the end, the average number of neighbors for each malicious link, which depends on the network density d is given by:

$$0.29\pi r^2 d \qquad (8)$$

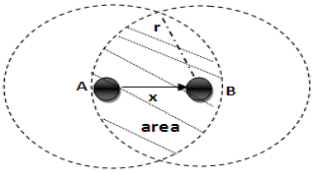

Fig. 3. Neighbors of a malicious link A-B

According to the works [19], an attacker is always encouraged to attack specifically when he detects relevant information, such encrypted messages. In this study, we take into account this fact and force the passive threat to act as an active one in order to lead IDS agents to detect it. Furthermore, we demonstrate in the following according to our simulation that within a cluster when a time's period *Td* is elapsed and none of active attacks are occurring, the passive attack switches to an active one. In other word, as stated before, this period is a required time of nodes within a same cluster to encrypt their exchanged messages before switching to a plaintext message. The time's period *Td* is computed as follows:

$$Td = \mathrm{m}0.29\pi r^2 d\delta \qquad (9)$$

Here, m is the average number of links at a cluster. In the following, we study the detection rate of our approach only under the occurrence of a passive attack by varying the value δ. In this case, as stated above, we fix the number of passive attacks equal to 30% of overall nodes. This study relies on the following assumption: the passive attack is mobile and can move from one place to another in order to hear the packets that traverse its radio range.

According to Fig. 4, when the value of δ increases (i.e exceed 80 milliseconds) the detection rate of passive attacks tends to 98 %. However, the optimal value of δ that suits the requirements of our application, i.e. high detection rate and low overhead is equal approximately to 20 milliseconds, in such case the detection rate is equal to 97%. In addition, according to our result, we found out that the time *Td* depends on the average number of neighbors of malicious links at each cluster. Therefore, when this number increases, the encryption time within a cluster becomes important. As a result, a tradeoff between the security level and energy consumption must be taken into account.

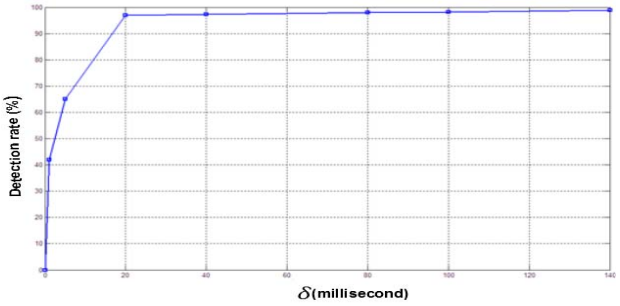

Fig. 4. Passive attacks' detection rate versus δ

At the end we conclude that, with the help of cryptography technique, the intrusion detection can remove almost all passive attacks from the network since such technique makes the information more attractive for the intruder to attack. In addition, even when the passive attack doesn't switch to an active one, such threats cannot overhear at all time the plaintext message as the keys are updated several time during the network lifetime and thus data confidentiality is assured.

C. Results and analysis

In this subsection, as stated above we evaluate the performance of LHSFW by compare it with hybrid security mechanisms proposed by S. Shin and all [2], and W.T. Su and all [3]. This study relies on the evaluation of the detection rate, false positive rate, average efficiency and energy consumption for each framework when the attacks cited above occur. In addition, we vary the number of attackers from 5 % to 50% of overall nodes.

Detection and false positive rates. According to Fig. 5 (a) and (b), we can see that LHSFW outperforms the hybrid security mechanisms proposed in current literature [2][3], in terms of attacks detection's accuracy, i.e. high detection and low false positive rates. Moreover, unlike these two security frameworks [2][3], LHSFW has the ability to detect passive attacks with a high accuracy by attracting such threat to switch to an active attack as proved above. .

Average Efficiency. LHSFW spends a short time compared to hybrid security mechanisms [2][3] (see Fig. 6(a)), specifically when the number of malicious nodes is important. This result occurred due to the fact that our framework relies on cooperative detection between the IDS agents and their respective CH to detect suspected nodes in a less time, unlike the frameworks [2][3] where there is no cooperation between IDS nodes, which leads them to spend a considerable amount of time to confirm the malicious behavior of a monitored node.

Energy Consumption. Sensors energy is a very important point in the design and implementation of the application. According to the Fig. 6 (b), it is apparent that LHSFW exhibits the lowest energy consumption to detect all attacks cited above compared to the hybrid security mechanisms[2][3], specifically when the number of nodes is important (equal to 50% of overall nodes). This result is achieved due to the fact that LHSFW aims to launch the cryptography operation only if necessary (as described above).

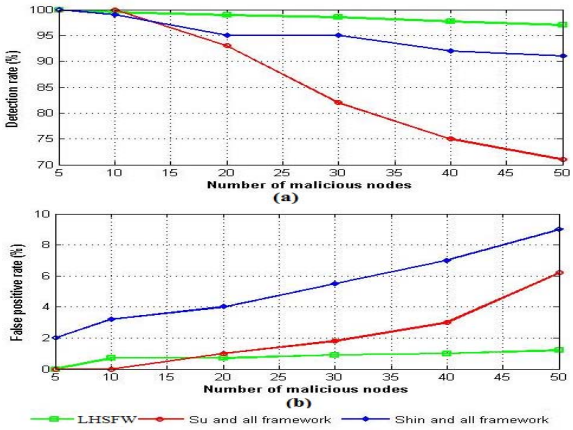

Fig. 5. LHSFW performances in terms of: (a) Detection rate, (b) False positive rate

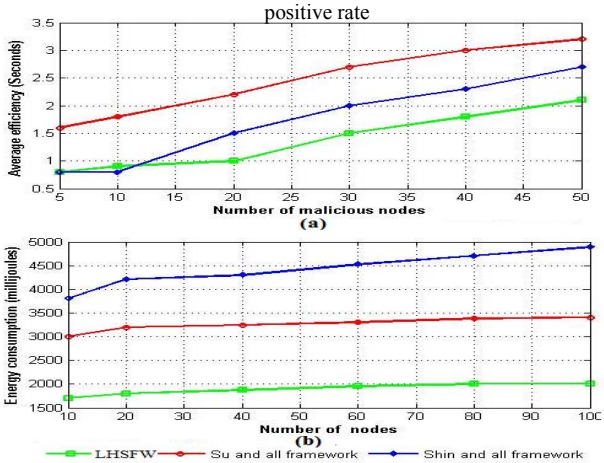

Fig. 6. LHSFW performances in terms of: (a) Average efficiency, and (b) Energy consumption

VI. CONCLUSION

In this paper, we develop a lightweight security framework for WSN, which aims to assure data confidentiality by using a cryptography technique and an intrusion detection approach to detect the internal attacks. However, such combination could generate a high overhead. In this research work, our aim is to address this issue by launching the cryptography operation only at the attacked clusters. According to our simulation results, when the active attacks occur, LHSFW framework outperforms other security frameworks in terms of detection rate, false positive rate, energy consumption, and efficiency. Furthermore, we prove that, by using cryptography technique our intrusion detection approach removes almost all passive attacks from the network as such technique makes the information more attractive for the intruder.

In the near future, we will take into account the context of mobility with embedding our framework on such network and will evaluate the delay and the amount of overhead generated in a scaling mode.

REFERENCES

[1] H. Kumarage, I. Khalil, Z. Tari, A. Zomaya, Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling, Journal of Parallel and Distributed Computing 73(6), 2013, pp. 790-806.

[2] S. Shin S, T. Kwon , G.Y. Jo , Y. Park , H. Rhy , An experimental study of hierarchical intrusion detection for wireless industrial sensor networks, IEEE Transactions on Industrial Informatics 6 (4), 2010, pp.744–757.

[3] W.-T. Su, K.-M. Chang, Y.-H. Kuo eHIP: An energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks , Computer Networks 51(4), 2007, pp. 1151-1168.

[4] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy efficient communication protocol for wireless microsensor networks, The 33rd International Conference on System Sciences, IEEE, Hawaii, USA, 2000, pp.1-10.

[5] H. Sedjelmaci, S.M. Senouci, M. Feham, An efficient intrusion detection framework in cluster-based wireless sensor networks, Security and Communication Networks, 6(10), 2013, pp. 1211–1224.

[6] A. H. Farooqi, F. A. Khan, Intrusion detection systems for wireless sensor networks: a survey, in Proc. Communications in Computer and Information Science, Springer ,vol. 56, 2009, pp. 234–241.

[7] G. Li, J. H, Y. Fu, Group-based intrusion detection system in wireless sensor networks, Computer Communications 31(18), 2008, pp. 4324–4332.

[8] S.K. Udgata, A. Mubeen. S.L. Sabat., Wireless sensor network security model using zero knowledge protocol, IEEE International Conference on Communications (ICC), 2011, pp. 1-5.

[9] S. Hussain, M. S. Rahman, Using received signal strength indicator to detect node replacement and replication attacks in wireless sensor networks, SPIE Proceedings on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, Orlando, USA, Vol. 7344, 2009.

[10] G. Han, W. Shen, T. Q. Duong, Mohsen Guizani and Takahiro Hara, A proposed security scheme against Denial of Service attacks in cluster-based wireless sensor networks, Security and Communication Networks 2012.

[11] B. Yahya, J. B. Othman: Towards a classification of energy aware MAC protocols for wireless sensor networks. Wireless Communications and Mobile Computing 9(12), 2009, pp. 1572-1607.

[12] O. Younis , S. Fahmy, HEED: a hybrid energy efficient distributed clustering approach for ad hoc sensor networks, IEEE Transactions on Mobile Computing 3(4), 2004, pp. 366–379.

[13] Y. Cheng, D. P. Agrawal, An improved key distribution mechanism for large-scale hierarchical wireless sensor networks, Ad Hoc Networks, 5(1), 2007, 35-48.

[14] NIST-CSRC, SKIPJACK and KEA algorithm specification, version 2, 1998. Available from: http://csrc.nist.gov/CryptoToolkit/.

[15] R. Roman, C. Alcaraz, J. Lopez, A Survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes, Mobile Networks and Applications 12(4), 2007, 231-244.

[16] Simulating tinyOS networks, 2003. Available at http://www.cs.berkeley.edu/ pal/research/tossim.html.

[17] Efficient power simulation for TinyOS applications, 2004. Available at: http://www.eecs.harvard.edu/shnayder/ptossim/.

[18] I. Khalil, S. Bagchi, N. B. Shroff, LITEWORP: A lightweight countermeasure for the wormhole attack in multi hop wireless networks, In Proc. International Conference on Dependable Systems and Networks, IEEE, Yokohama, Japan, 2005, pp.612-621.

[19] A. Agah , S.K . Das, K. Basu , A non-cooperative game approach for intrusion detection in sensor networks, IEEE Vehicular Technology Conference ,Los Angeles, USA,2004, pp.2902 - 2906.