Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 348606, 7 pages
http://dx.doi.org/10.1155/2014/348606

Research Article

Web Spider Defense Technique in Wireless Sensor Networks

Alejandro Canovas,[1] Jaime Lloret,[1] Elsa Macias,[2] and Alvaro Suarez[2]

[1] *Integrated Management Coastal Research Institute, Universidad Politécnica de Valencia, Spain*
[2] *Departamento de Ingeniería Telemática, Universidad de Las Palmas de Gran Canaria, Spain*

Correspondence should be addressed to Alejandro Canovas; alcasol@posgrado.upv.es

Received 22 April 2014; Accepted 1 July 2014; Published 23 July 2014

Academic Editor: S. Khan

Wireless sensor networks (WSNs) are currently widely used in many environments. Some of them gather many critical data, which should be protected from intruders. Generally, when an intruder is detected in the WSN, its connection is immediately stopped. But this way does not let the network administrator gather information about the attacker and/or its purposes. In this paper, we present a bioinspired system that uses the procedure taken by the web spider when it wants to catch its prey. We will explain how all steps performed by the web spider are included in our system and we will detail the algorithm and protocol procedure. A real test bench has been implemented in order to validate our system. It shows the performance for different response times, the CPU and RAM consumption, and the average and maximum values for *ping* and *tracert* time responses using constant delay and exponential jitter.

1. Introduction

A wireless sensor network (WSN) is distributed in nature. It consists of several electronic devices with a memory, a processor, and one or more elements that sense the environment [1, 2]. One of their main issues taken into account in their deployment is their power limitation and their need to save energy [3, 4]. Sensor nodes can communicate among them using a particular or standard communication technology network interface card. The sensed values can be forwarded to a central manager that usually is a computer (or similar device). The computer allocates a manager that is in charge to manage the WSN. The most common strategy to read values of sensed elements consists of interrogating the manager in order to obtain a set of sensed values. In this sense, a WSN is used to collect and monitor the related information about a specific environment. This procedure has relevance in several cases: vigilance, oceanographic values of a strategic installation, police related information, and many more.

Generally, WSNs are used to sense private data. Some of them can also transmit critical data. Thus, it is very important to secure the collection of data and detect and avoid external intrusions. An intruder may be able to access unauthorized data, spread erroneous data and/or malicious code, implement unauthorized changes to data or sensor software, or steal data. Moreover, an intruder could initiate attacks to the network from that sensor node and open new doors to other intruders.

This must be done taking into consideration four requirements: data confidentiality, data authentication, data integrity, and denial-of-service (DoS) attack avoidance [5]. Different surveys on WSN security are presented in [6–13]. In [14, 15] a list of attacks and counterattacks are surveyed.

This work is focused on intrusion attacks in WSNs. A network intrusion detection system (IDS) is an essential element in a computer security strategy [16]. An IDS is a device or a software application that monitors network and system activities for malicious activities or policy violations. The IDS produces reports to a central system that allow humans to intervene or that can be responded by computer systems in an attempt to stop the intrusion. In a WSN, this attack means that an attacker (malicious user) wants to illegally read the data sensed by a set of sensors. We suppose the malicious user can interrogate the sensors in the WSN bypassing the control of the WSN manager/administrator. The difficult task here is to discover when a malicious attack is

happening and which the particular properties of the attacker are. The main idea behind this is that the IDS can learn about the attacker in order to prevent future attacks. Several techniques have been used to design the IDS.

(i) The intrusion detection policy proposed in [17] monitors the communication between neighboring nodes and finds those nodes that are not working normally. Some general rules are defined to detect such nodes, which are called compromised nodes. They simulated transport and routing layer in order to analyze the performance of the proposed policy. They showed that each node should be treated independently in the WSN, and purely centralized detection schemes may fail to identify the network behavior whether it is normal or it is under any attack.

(ii) Due to the huge volume of network traffic, coding the rules becomes difficult and time-consuming. Data mining techniques, used for example, in anomaly based systems [18], can build network intrusion detection models adaptively. They can analyze and predict the behaviors of users in order to know if these behaviors are attacks or a normal behavior.

(iii) The traffic prediction can also be used to model a mechanism against any intrusion detection. In [19] it is shown that, by inspecting received packet features, a sensor can identify an intruder impersonating a legitimate neighbor.

(iv) A honeypot is usually a valuable surveillance tool that provides early warnings to system administrator about the trends of malicious activity in the WSN. A wireless honeypot can be used to gather information about the intruder in the WSN, taking into account several implementation techniques for wireless local area network [20]. In a WSN a fake access point could be implemented by a sensor that responds with fake data to the intruder. A very interesting survey that includes results of honeypot technology applied to WSN can be found in [21]. The information sensitivity, resources, and time are the most important factors in choosing the type of honeypot for any WSN. We differentiate two types of honeypots: (a) low-interaction, which only monitors for anomalies, and (b) high-interaction, where detailed information of the requests is used for predicting future attacks using pattern recognition. A multilevel security defense is presented in [22], which considers a hierarchical WSN. The authors arrange regular sensors, gateways that are in charge to control regular sensors, base stations that control the gateways, and honeypots that collaborate with base stations. In each level a different kind of attack can be controlled.

(v) Artificial intelligent based mechanisms: exploiting knowledge about the nature of biological systems can result in valuable information about the attacker. For example, bioinspired solutions are applied to efficient computing (bioinspired computing), making robots that are inspired by the biological systems (bioinspired robotics), technical developments in engineering (bioinspired systems), and networking (bioinspired networking). So, they can also be applied to the design of an IDS for WSN. Honeypot can be considered an artificial intelligent technique due to the fact that it mimics the biological nature of particular species. Artificial intelligence is becoming an effective method to be applied in security detection systems [23].

This work centers our attention on artificial bioinspired security mechanisms for IDS in WSNs. We have designed an algorithm and a protocol to detect an intrusion attack inspired in the web spider behavior when an attack suffered in its web [24]. We technically implement our algorithm and protocol considering the honeypots technique. In contrast to [17], we are not concerned with routing inside the WSN, but in addition to that work we propose a transport and policy algorithm and protocol. We do not inspect the traffic of the intruder (as [19] did) but we consider it to reduce the rate of attacks it can do. Our objective is to gain time to find out information about the intruder. To do this, we implement a low-interaction sensors honeypot that tries to detect the intruders and then delays the answer to them for earlier learning of their future behavior. In contrast to [22], we do not consider a hierarchical WSN. We consider all the nodes are regular and have the same role in the network (honeypot sensors and real sensors).

The paper is organized as follows. In Section 2, we analyze the works found about bioinspired mechanisms used in security. Section 3 describes our web spider-inspired proposal. The system algorithm and protocol are explained in Section 4. Test bench experiments and results are included in Section 5. Finally, Section 6 draws the conclusion and future work.

2. Related Work

The section shows some works related to bioinspired mechanisms for security in WSN.

A survey on practical applications and open research issues for bioinspired self-organized networking (SON) systems is presented in [25]. The benefits of using these bioinspired techniques against conventional SON solutions include, but are not limited to, lower MAC delays, communications overhead and hardware complexity, higher adaptivity to changes, and resource utilization. Considering the benefits of these techniques, SON systems, such as WSN and wireless ad hoc networks, can exploit the improvements introduced by the bioinspired techniques compared to the isolated conventional SON solutions.

The authors in [26] apply the biological knowledge about the human immune system to propose a new network security mechanism to disable the fraudulent nodes in a WSN. Bioinspired algorithms provide dynamic, adaptive, and real-time methods of intrusion detection. The work included in [27] presents a review on genetic algorithm, artificial immune, and artificial neural network (ANN) based intrusion detection systems (IDS) techniques used in WSN.

Moreover, an algorithm inspired on the human immune system behavior to detect intruders in WSN is presented in [28].

A key component of bioinspired response methods is the use of feedback from the network to better adapt their response to the specific attack [29]. The author developed a method to calculate response times for a WSN that could be used to improve the bioinspired method for selecting the most suitable intrusion response for ad hoc networks.

In [30], a honeypot based framework is proposed that is used to earlier learn future attacks of the intruder and serve as a defensive countermeasure. It is based on the biological behavior of a particular species of ant. The ants store food forming a living repository of food and are often attacked by raiders. They considered a WSN as composed by two types of ants: honey ants and real ants. They strategically distribute the honeypot sensors (honey ants) that will mimic the physical data (real ants). Then, the IDS will induce traffic from alleged intruders to these honeypot sensors. This is done by implementing a swarm intelligence algorithm that takes into account the communication among sensors like the ants do. They route virtual values to confuse the intruder and also to make it believe that it is receiving real values. In this way the intruder could be discovered earlier.

Most bioinspired methods for WSN intrusion attacks are generally applied to a single protocol layer of the OSI stack, for example, (i) genetic algorithm at the physical layer; (ii) antiphase synchronization at the MAC layer, a bioinspired method based on the behavior of Japanese tree frogs; (iii) ant colony optimization at the network layer; (iv) and quantified trust models at the application layer. At present the combination of several bioinspired methods for WSN is applied to improve the system performance [31].

We propose a honeypot implementation for IDS in a WSN, which is bioinspired in the behavior of the web spider. We have only found one paper that uses a web spider-inspired mechanism [32]. It proposes a bioinspired algorithm based on the social behavior of spiders from Congo to detect and eliminate misbehaving sensor nodes in WSN. The biological inspiration comes from the fact that these kinds of spiders form a collaborative group to listen vibrations of victims in the web in order to hunt them. The bioinspired algorithm is distributed among sensor nodes (spiders) and it works as follows: one or more sensor nodes detect an attack from a suspected node (victim); then the sensor node sets a first level of alert and sends this detection to all their neighbors (collaboration); to reduce false alarms in the detection, the algorithm sets that if a second attack from the same suspected node is detected for the same sensor that detected the first attack or for a neighbor sensor, then the suspected node is considered as an intruder node. The paper does not present how and why this node is considered suspicious and how to reduce this intruder.

As far as we know there is not any other work published that uses the web spider behavior for WSN security. Moreover, the work presented in this paper is completely different from [32]. We have used different parts of the web spider behavior than the ones presented in [32].

3. Web Spider Defense Description

This section presents the description of the web spider defense technique and how it is applied to our system.

Spiders are often underestimated as suitable behavioral models. Spiders show surprising cognitive abilities, changing their behavior to suit their situational needs [33]. All spiders are predators. There are many types of spiders and there is a wide variety of methods used by them to capture their prey. Some spiders are hunters that chase and overpower their prey. Other spiders instead weave silk snares, or webs, to capture their prey [34, 35]. Some spiders inject poison into their prey. The poison paralyzes victims making them lose mobility. After paralyzing victims, spiders usually wrap their victims with silk and soften the meat with gastric juice. Finally spider absorbs the result of this mixture. The behavior that we are going to use in our system is the behavior of web spiders that use poison to paralyze their prey once it is trapped in the web. There are several types of web spiders, which can be spiral orb web, tangle web or cobweb, funnel web, tubular web, and sheet web.

When a spider wants to capture a prey, it builds a web and waits till some flies or mosquitoes are trapped in it. When it happens, the spider has a delicacy to attack bigger preys. It has just to wait some time till a new prey sees the fly and/or the mosquito and gets trapped when it tries to catch them. Now the procedure to paralyze this big prey is injecting poison, which slows down the mobility of the prey till it has no mobility.

This procedure is used by our system. We will use one or several fake wireless sensor nodes placed in the WSN, which announce network services and provide false data. These nodes have few or no security. It (or they) will be honeypots for the intruders. The idea of attracting attackers is not really new. It has been used in many other types of networks [36]. As soon as the fake wireless sensor node detects a connection, it will contact the network administrator, which will follow the connection and gather information from the intruder (such as getting the IP address and DNS name). Fake wireless sensor nodes, where security level is very low, will detect intruders by using any of the existing intrusion detection systems [11]. They will send data to sink nodes as regular nodes, but these fake data will be discarded by the sink node. In order to keep the intruder busy, the fake wireless sensor node slows down the replies to the intruder messages, like the poison of the spider when the prey is trapped in the web.

The system uses the connection establishments to keep intruders trapped. Every request is replied before the timeout, but it is delayed in order to let the system administrator gather information about the intruder. The system administrator is a node that is placed in the network, whose purpose is to gather information about a node through its IP address, DNS name, traces, and so forth.

4. System Algorithm and Protocol

This section presents the algorithm designed for our system and the protocol created for the proper operation of our system.

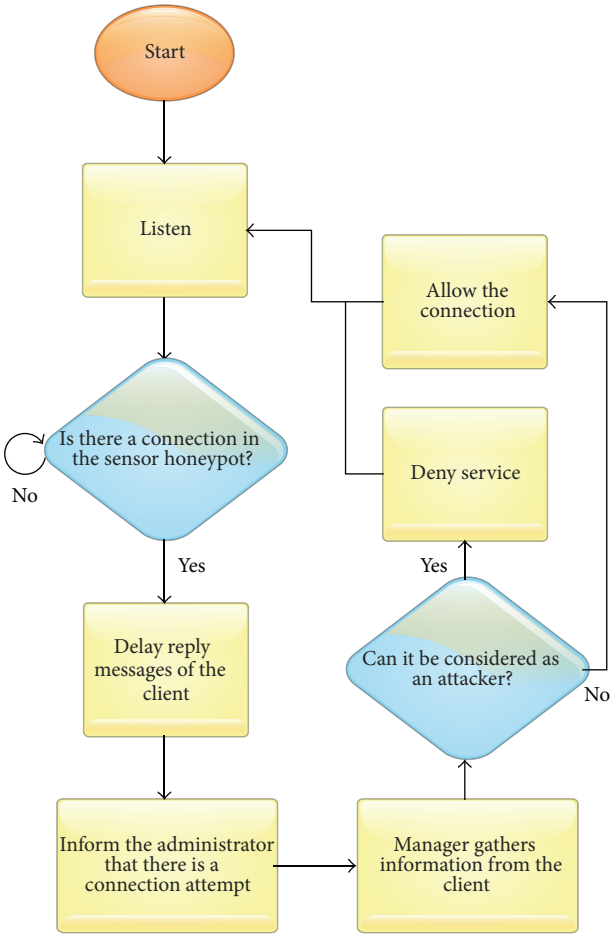

Figure 1: Spider defense algorithm.

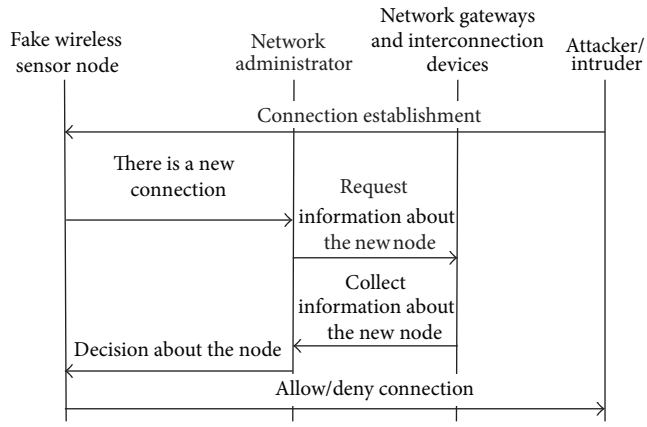

Figure 2: Network protocol.

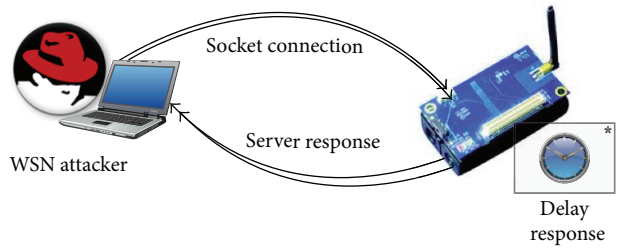

Figure 3: System architecture for the 1st experiment.

Figure 1 shows the algorithm used for the intrusion or attackers detection and the steps followed to slow down their connections. At the beginning, the system listens if the fake wireless sensor node is receiving any connection request. If it receives a request, it slows down the connection and informs the network administrator that it has a possible intruder. This slow process is performed by a "wait procedure," which delays the replies. The delay time is lower than the threshold used by TPC connections for the exceeded time. These delays in the replies allow the network administrator to gather information about the intruder in order to identify it. The network administrator will be able to use any information gathering technique using echo request/reply, who is, and so forth. This information will be used to know if the user establishing the connection is an intruder or an attacker. If system confirms that the user is an intruder or an attacker, it will deny the service. If the user has the rights to perform this task because it belongs to the system, then the connection is established correctly and it goes to the listen state.

The designed protocol is shown in Figure 2. When the fake wireless sensor node receives a connection, it first sends a message to the network administrator in order to ask whether it is a trustable node or an intruder/attacker. Meanwhile it slows down the connection. The network administrator requests information about that node (by using its IP address, DNS name, traces, etc.) to the network gateways and interconnection devices. It gathers the information received about the type of node establishing the connection and informs the fake wireless sensor node. Then, it takes the appropriate action by denying or accepting the connection.

5. Test Bench Experiments

In order to carry out the performance study two experiments have been made. In both cases, the WSN attacker acts as the *client* and the wireless sensor node as *server*. Both communicate using TCP sockets and the communication is established following a three-way handshake algorithm. The last answer (segment [FIN, ACK]) is delayed to give time to the network manager to diagnose the connection as a secure or insecure one.

5.1. Experiment 1. Figure 3 shows the system architecture used for the first experiment. The WSN attacker uses a MacBook Pro with the following characteristics: Intel Core 2 Duo 2.4 GHz processor and 2 GB RAM. The sensor node has a 1.6 GHz processor and 1 GB RAM. The communication between the WSN attacker and sensor node is wireless. Wireshark sniffer program running in the WSN attacker node is used to compute the elapsed reply time. Both, client and server programs, have been coded in Java programming language.

Figure 4 shows different response times from the wireless sensor node to the WSN attacker according to the artificial

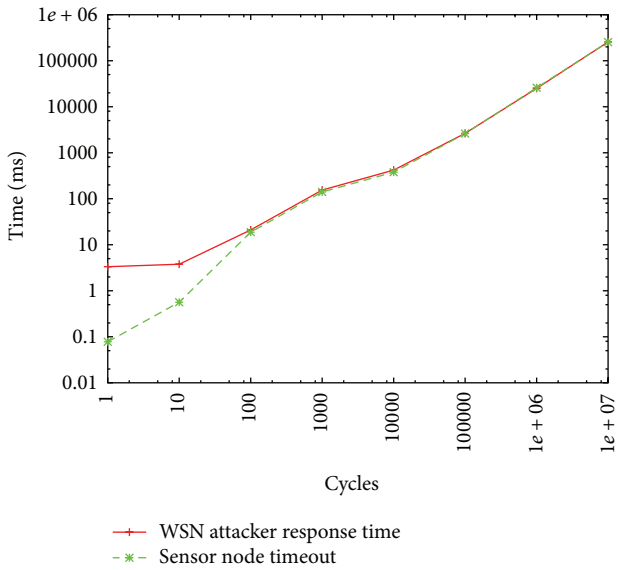

FIGURE 4: Response time as a function of the number of cycles.

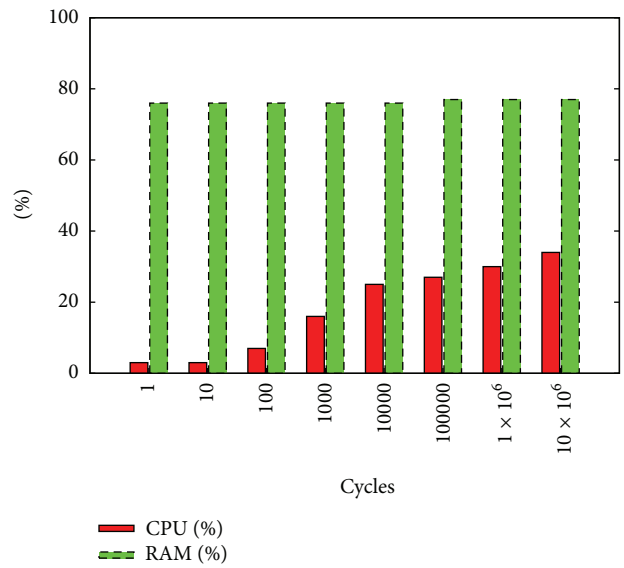

FIGURE 5: CPU and RAM usage as a function of the number of cycles.

delay introduced by the sensor node (a loop varying the response time generates these different response times). As you can see, if the delay is high, the total amount of time elapsed from the first segment [SEQ] to the reception of the last segment [FIN, ACK] is closer to the artificial delay. This is not true if the artficial delay is low.

Figure 5 shows the CPU and RAM consumption during the loop execution. The measurements were obtained with *top* Linux program. As it can be seen, RAM usage is not high enough to be considerable. On the contrary, the more the delay in the sensor node, the higher the CPU consumption in the sensor node. The delay should be close to the time needed by the network manager to diagnose if the attempt of connection initiated by the WSN attacker is secure or not. Moreover, we have to look for the minimum delay value that will affect the system performance, which is why we performed the second experiment.

5.2. Experiment 2. This second experiment helps us to determine the delay by measuring the reply time of the *tracert* and *ping* to the wireless sensor nodes in a network with different delays.

Figure 6 shows the system architecture used for the second experiment. The WSN attacker and each one of the 12 sensor nodes use the same equipment described for the previous experiment (the attacker uses a MacBook Pro with Intel Core 2 Duo 2.4 GHz processor and 2 GB RAM and the sensor node with 1.6 GHz processor and 1 GB RAM). Again, the communication between the WSN attacker and sensor node is wireless and the Wireshark sniffer program is running in the WSN attacker to compute the elapsed reply time.

Each sensor node is accessible from the WSN attacker via Internet. *NetDisturb* program [37] let us vary several network parameters such as the delay and jitter. Next we present the obtained simulation results.

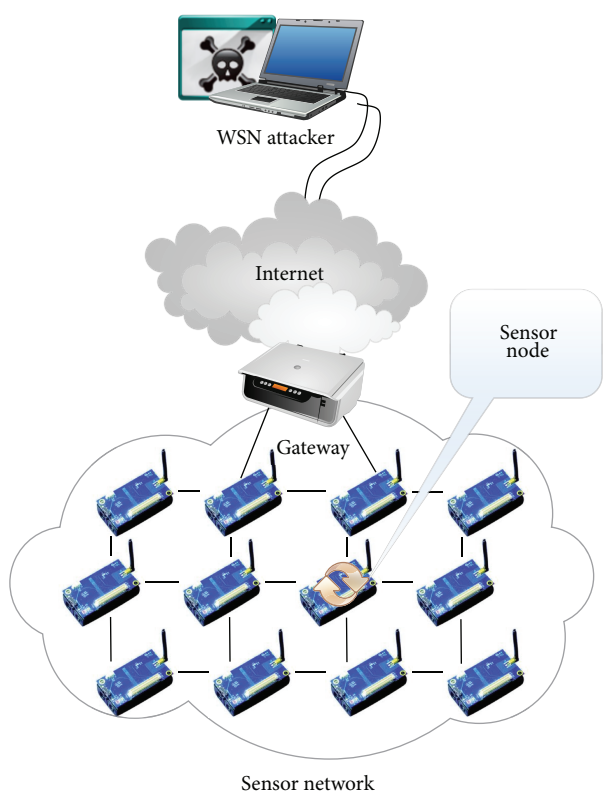

FIGURE 6: System architecture for the 2nd experiment.

Figure 7 shows both, average and maximum values for *ping* and *tracert* time responses for different constant network delays.

As Figure 7 shows, the more the network delay, the higher the response time for *ping* and *tracert*. An important issue derived from our experimentation is that the probability of

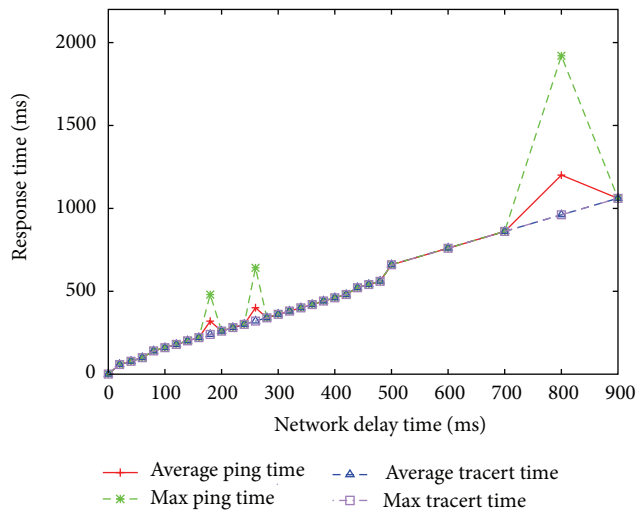

Figure 7: Response time for the *pings* and *tracerts* in the WSN with constant delay.

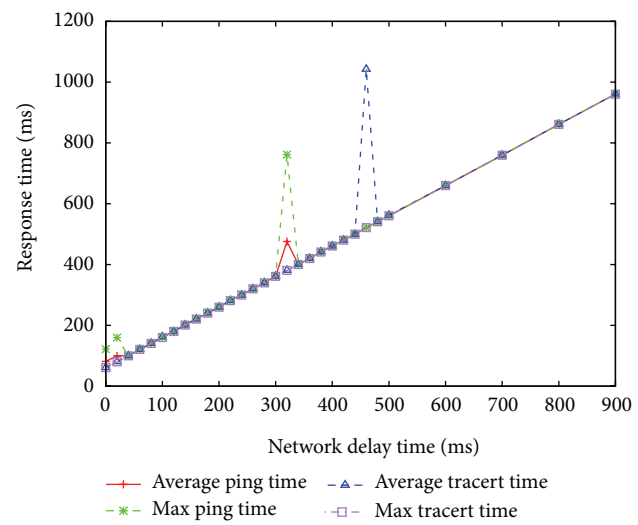

Figure 8: Response time for the *pings* and *tracerts* in the WSN with exponential jitter.

having a peak is higher for high network delays. Figure 7 demonstrates that this probability increases from a network delay higher than 200 ms. From this figure, we can make an estimation of the amount of time needed by the network manager to give a diagnosis about the connection between the WSN attacker and wireless sensor node. For example, if the network delay is 100 ms, the network manager takes into account the fact that the response time is 160 ms on average. As a result, the time to answer to the WSN attacker connection request should be greater than 160 ms.

Figure 8 shows results obtained varying exponentially the jitter according to the following equation:

$$f(x) = \lambda e^{-\lambda x} dx \quad \text{if } x \geq 0,$$
$$f(x) = 0 \quad \text{if } x < 0, \tag{1}$$

where $\lambda = 10$ and x is the delay variation.

As Figure 8 shows, there is higher probability to obtain a peak using *tracerts*. Another observation is that *ping* and *tracert* behavior is lineal in this experiment in comparison with Figure 7. The lineal behavior assists the network manager to predict the response time.

6. Conclusion

In this paper, we have presented a bioinspired system that uses the web spider hunting technique. We have explained how all steps performed by the web spider are included in our system. Moreover, we have detailed the system algorithm and the protocol procedure for the proper operation of the system. A real test bench has been implemented in order to validate our system.

In order to carry out our performance study, we have made two experiments. First, we tested performance of the direct communication between the WSN attacker and the wireless sensor node. Then, we performed a second experiment to measure the reply time of the wireless sensor nodes in a network with different delays.

In future works we will make performance experiments using one and several wireless attackers in order to know response times for the *ping* and the *tracert*. Moreover, our system will include other spider behaviors from other types of spiders. Now we are developing the system for a real environment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work has been partially supported by the "Ministerio de Ciencia e Innovación", through the "Plan Nacional de I+D+i 2008–2011" in the "Subprograma de Proyectos de Investigación Fundamental", Project TEC2011-27516.

References

[1] D. Bri, M. Garcia, J. Lloret, and P. Dini, "Real deployments of wireless sensor networks," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications (Sensorcomm '09)*, pp. 415–423, Greece, Athens, Ga, USA, June 2009.

[2] M. Garcia, D. Bri, S. Sendra, and J. Lloret, "Practical deployments of wireless sensor networks: a survey," *Journal On Advances in Networks and Services*, vol. 3, no. 1-2, pp. 170–185, 2010.

[3] S. Sendra, J. Lloret, M. García, and J. F. Toledo, "Power saving and energy optimization techniques for wireless sensor networks," *Journal of Communications*, vol. 6, no. 6, pp. 439–459, 2011.

[4] M. Segal, "Improving lifetime of wireless sensor networks," *Network Protocols and Algorithms*, vol. 1, no. 2, pp. 48–60, 2009.

[5] S. Kuncha and P. V. G. D. P. Reddy, "Impact of security attacks on a new security protocol for mobile ad hoc networks," *Network Protocols and Algorithms*, vol. 3, no. 4, pp. 122–1403, 2011.

[6] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1302–1325, 2011.

[7] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: attack analysis and countermeasures," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 867–880, 2012.

[8] W. T. Zhu, J. Zhou, R. H. Deng, and F. Bao, "Detecting node replication attacks in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 1022–1034, 2012.

[9] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *International Journal of Computer Science and Information Security*, vol. 4, no. 1-2, 2009.

[10] Y. Maleh and A. Ezzati, "A review of security attacks and intrusion detection schemes in wireless sensor network," *International Journal of Wireless & Mobile Networks*, vol. 5, no. 6, pp. 1–12, 2013.

[11] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: a review," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 167575, 7 pages, 2013.

[12] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 56–63, 2007.

[13] H. C. Chaudhari and L. U. Kadam, "Wireless sensor networks: security, attacks and challenges," *International Journal of Networking*, vol. 1, no. 1, pp. 4–16, 2011.

[14] N. Fatema and R. Brad, "Attacks and counterattacks on wireless sensor networks," *International Journal of Ad hoc, Sensor & Ubiquitous Computing*, vol. 4, no. 6, pp. 1–15, 2013.

[15] A. Radhika, D. Kavitha, and D. Haritha, "Mobile agent based routing in MANETs—attacks & defences," *Network Protocols and Algorithms*, vol. 3, no. 4, pp. 108–121, 2011.

[16] R. H. Jacobsen, Q. Zhang, and T. S. Toftegaard, "Bioinspired principles for large-scale networked sensor systems: an overview," *Sensors*, vol. 11, no. 4, pp. 4137–4151, 2011.

[17] J. Xu, J. Wang, S. Xie, W. Chen, and J. Kim, "Study on intrusion detection policy for wireless sensor networks," *International Journal of Security and its Applications*, vol. 7, no. 1, pp. 1–6, 2013.

[18] M. S. Sisodia and V. Raghuwanshi, "Anomaly base network intrusion detection by using random decision tree and random projection: a fast network intrusion detection technique," *Network Protocols and Algorithms*, vol. 3, no. 4, pp. 93–107, 2011.

[19] H. Zhijie and W. Ruchuang, "Intrusion detection for wireless sensor network based on traffic prediction model," *Physics Procedia*, vol. 25, pp. 2072–2080, 2012.

[20] N. Al-Gharabally, N. El-Sayed, S. Al-Mulla, and I. Ahmad, "Wireless honeypots: survey and assessment," in *Proceedings of the Conference on Information Science, Technology and Applications (ISTA '09)*, pp. 45–52, ACM, March 2009.

[21] V. Gopinath, *Success analysis of deception in wireless sensor networks [M.S. thesis]*, Oklahoma State University, 2010.

[22] S. K. Srivastava, B. K. Mishra, and B. K. Mishra, "Security framework against malicious attacks in wireless sensor network," *International Journal of Advanced Technology & Engineering Research*, vol. 3, no. 5, pp. 7–11, 2013.

[23] N. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Artificial neural network based detection of energy exhaustion attacks in wireless sensor networks capable of energy harvesting," *Journal of Ad Hoc & Sensor Wireless Networks*, vol. 2013, pp. 1–25, 2013.

[24] "Mecanismos de defensa de las arañas," http://www.aracnipedia.com/mecanismos-defensa-aranas/.

[25] Z. Zhang, K. Long, J. Wang, and F. Dressler, "On swarm intelligence inspired self-organized networking: its bionic mechanisms, designing principles and optimization approaches," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 513–537, 2014.

[26] H. Rathore and S. Jha, "Bio-inspired machine learning based wireless sensor network security," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC '13)*, pp. 140–146, Fargo, ND, USA, August 2013.

[27] N. A. Alrajeh and J. Lloret, "Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 351047, 6 pages, 2013.

[28] J. Baburajan, "Intrusion detection in wireless sensor networks using watchdog based clonal selection algorithm," *International Journal of Research in Engineering & Advanced Technology*, vol. 1, no. 1, 2013.

[29] M. K. Amirkolaei, *Enhancing bio-inspired intrusion response in Ad-hoc networks [Ph.D. thesis]*, Edinburgh Napier University, Edinburgh, UK, August 2013, http://researchrepository.napier.ac.uk/6533/.

[30] R. Muraleedharan and L. A. Osadciw, "An intrusion detection framework for sensor networks using Honeypot and Swarm Intelligence," in *Proceedings of the 6th Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '09)*, Toronto, Canada, July 2009.

[31] W. S. Hortos, "Bio-inspired, cross-layer protocol design for intrusion detection and identification in wireless sensor networks," in *Proceeding of the 37th IEEE Conference on Local Computer Networks Workshops, LCN Workshops*, pp. 1030–1037, Clearwater, Fla, USA, October 2012.

[32] K. Benahmed, M. Merabti, and H. Haffaf, "Inspired social spider behavior for secure wireless sensor networks," *International Journal of Mobile Computing and Multimedia Communications*, vol. 4, no. 4, pp. 1–10, 2012.

[33] M. E. Herberstein, *Spider Behaviour: Flexibility and Versatility*, Cambridge University Press, Cambridge, UK, 2011.

[34] Spiderword Website, *Spider Methods of Capturing Prey*, 2014, http://www.spidersworlds.com/spider-methods-of-capturing-prey/.

[35] R. F. Foelix, *Biology of Spiders*, Oxford University Press, 3rd edition, 2010.

[36] M. Ficco, "Achieving security by intrusion-tolerance based on event correlation," *Network Protocols and Algorithms*, vol. 2, no. 3, pp. 70–84, 2010.

[37] NetDisturb website, http://www.zti-communications.com/netdisturb.