Securing Wireless Sensor Networks from Denial-of-Service Attacks using Artificial Intelligence and the CLIPS Expert System Tool

Vincent F. Taylor
Faculty of Science and Technology
University of the West Indies
Mona, Kingston 7, Jamaica
Email: vincent.taylor@mymona.uwi.edu

Daniel T. Fokum
Faculty of Science and Technology
University of the West Indies
Mona, Kingston 7, Jamaica
Email: daniel.fokum@uwimona.edu.jm

***Abstract*—Wireless sensor networks consist of a number of autonomous sensor nodes which are deployed in various areas of interest to collect data and cooperatively transmit that data back to a base station. Wireless sensor networks have been used in military applications, environmental monitoring applications, healthcare applications, and even home applications. An adversary may want to disrupt these sensor networks for various reasons. Adversaries range from a hacker with a laptop to corporations and governments who have a vested interest in compromising the proper operation of an unwelcome sensor network. Since sensor nodes are small and usually placed in uncontrolled environments, they are susceptible to capture and reprogramming by an adversary. The low-power nature of sensor nodes make traditional strong encryption approaches to network security infeasible as nodes have limited processing power and sometimes significant energy constraints. This paper presents work in progress on developing a system which would protect a wireless sensor network from denial-of-service attacks after one or more nodes on the network have been captured and reprogrammed by an adversary. This system removes the need to rely on tamper proof packaging to protect the cryptographic keys and other sensitive data which is stored on nodes. With the proposed system, even if cryptographic keys are obtained by an attacker and are used to send false routing information or other spurious control information, the network will be able to identify such malicious nodes by using artificial intelligence and an expert system developed using the C Language Integrated Production System tool.**

I. Introduction

Sensor networks consist of geographically distributed sensor nodes which monitor physical or environmental characteristics, and cooperatively pass data to a base station. Wireless sensor networks were inspired by military applications such as border control but have now become pervasive and are being used in the health industry, in the environment, as well as in home applications [1]. The low-cost, low-power nature of sensor networks make security of these networks an interesting challenge as nodes have limited power supplies and processing power which makes the usual public-key cryptographic approach infeasible [2].

Wireless sensor networks usually exist in dangerous or inaccessible areas such as behind enemy lines or other environments with adverse environmental conditions, for example: undersea, near active volcanoes, or out in the wilderness [3]. These areas of deployment usually put the wireless sensor network out of the immediate reach of its operator, which further adds to the complexity of securing it. Because these networks are usually out in the open, nodes are susceptible to capture and reprogramming by an adversary [4]. Adversaries may choose to disrupt sensor networks for military advantage, economic gain, or simply for the challenge. They may capture a node and reprogram it and/or steal sensitive data such as cryptographic keys from it. If these reprogrammed nodes are then inserted back into the sensor network, they may be used to cause mayhem such as jamming, collisions, resource exhaustion, unfairness, greed, misdirection, black holes and flooding [5]. A comprehensive solution needs to be found to mitigate the effects of successful node capture and reprogramming by an adversary.

The most vexing concern with sensor networks is their susceptibility to node capture. A comprehensive solution needs to be found to mitigate the effects of successful node capture and reprogramming by an adversary. It is prudent to assume that if some nodes are out in the environment, they may be captured and their cryptographic keys will be recovered by an adversary. Once this sensitive data is in the hands of an adversary, they can then proceed to modify the firmware of the captured node and reinsert it into the network to spoof messages, disobey protocol, and so on, all in an effort to disrupt the network [6]. A potential solution to the problem of node capture and data manipulation is the use of tamper-proof packaging, however, using tamper-proof packaging on motes may not be economically feasible [7].

Wireless sensor networks show much promise of providing low-cost monitoring solutions for international borders, the health of the elderly, and battlefields and will undoubtedly proliferate in the years to come. But security of these networks can no longer be an afterthought. It would be unwise to wait until the technology is ubiquitous to address the problems outlined above. A comprehensive solution needs to be developed to actively protect wireless sensor networks from denial-of-service attacks by adversaries.

In this paper, we propose a system called ADIOS (Advanced Detection of Intrusions On Sensor networks). This is a novel expert-system based intrusion detection and prevention system developed using the C Language Integrated Production

978-1-4799-0053-4/13/$31.00

System (CLIPS) expert system building tool [8]. ADIOS is designed to mitigate denial-of-service attacks in wireless sensor networks by capturing and analysing network events in a resource-friendly manner.

The rest of this paper is organized as follows: Section II gives background information on the problem as well as a review of the relevant literature. Section III describes using expert systems to secure against black hole attacks in wireless sensor networks. Section IV analyses how to integrate the suggested expert system with low-power motes in a resource-friendly manner. Finally, Section V concludes the paper by presenting future work and reporting some preliminary findings of the research.

II. Background

Network security consists of a set of principles and practices which are employed by an administrator to prevent and/or monitor unauthorized access to network-accessible resources [9]. Network security is of paramount importance since much personal and private data as well as financial transactions traverse computer networks. Adversaries, malicious individuals, and rogue governments have a vested interest in disrupting the networks of others and thus network administrators have an important task of protecting their networks.

As networks and computers become more pervasive, research into their security must come to the fore and integrated defence mechanisms for their protection must be studied. What we need are smart systems that are capable of identifying attacks as they are happening and proactively defending against these attacks while at the same time alerting administrators so that more permanent action can be taken against an intruder.

One means of network security is the intrusion detection and prevention system. An intrusion detection and prevention system works by monitoring pre-defined parameters of a network to check for known attack signatures, policy violations or other anomalies. If any interesting event is noticed, the system will usually generate an alarm or even take corrective measures to mitigate a potential attack [10]. The intrusion detection system that this paper describes utilises a combination of artificial intelligence, majority voting and key revocation.

A. Artificial Intelligence and Expert Systems

Artificial intelligence is a branch of computer science that aims to give machines the ability to perform simulated reasoning [11]. Given that wireless sensor networks and the routing protocols that run on them have certain expected behaviours under normal operating conditions, it is conceivable that artificial intelligence can be used to analyse network events in an effort to determine if there is any suspicious activity happening on the network [12].

An expert system being fed with network events may potentially be used as an intrusion detection system that can be used to protect a wireless sensor network. If expected network events or node behaviours are captured in the rule-base of an expert system, its inference engine can be used to determine if there are any anomalies. After encountering anomalies, ADIOS has the capacity to communicate with a majority voting mechanism in the wireless sensor network in order to seek to punish a node if neighbours determine that it seems to be misbehaving. Punishment of a node may take the form of key revocation, malice, or other suitable punishment and may be permanent, or have a duration that is some function of the severity of the misbehaviour.

B. Majority Voting

Majority voting systems are useful as they form a system of peer-review. This peer-review process would make it difficult for captured nodes to disrupt the operation of the wireless sensor network as all nodes would be checking, for example, for inconsistencies in routing states in the network [13]. This approach works for identifying nodes which may be attempting to manipulate routing information, but may not necessarily protect against other forms of attacks which do not affect the routing layer and thus a more comprehensive solution needs to be identified and developed.

For majority voting systems to work, the expectation is that each node will have more than one neighbour, and these neighbours will determine if their neighbours are behaving appropriately. This makes for a robust system which would be able to detect malicious nodes on the network [14].

Majority voting systems for wireless sensor networks have already been proposed [14], but none of these systems are invoked using a CLIPS-based intrusion detection system that identifies anomalies in node behaviour. The expert system based approach that ADIOS uses makes for a more robust system, as the definitions of appropriate node behaviour can be constantly improved and upgraded over time or modified depending on the particular sensor network application or threat scenario.

C. Key Revocation

Once a malicious node has been identified, by using ADIOS for example, its cryptographic keys can then be revoked, thereby preventing it from participating in the network [15]. The keys that will be potentially revoked may be the same cryptographic keys that were recovered by the adversary when the node was captured and before it was reprogrammed to become malicious. These network keys may have been used by the node to identify itself to the other nodes in the wireless sensor network. Thus, if the keys are in the hands of an adversary, there would be no way of readily knowing if the node was compromised and disseminating spurious information; hence the need for a more comprehensive method of identifying a malicious node.

Key revocation would effectively isolate the misbehaving node and render it ineffective at causing any more damage to the network. Combined with majority voting, key revocation can be remarkably useful for maintaining the integrity of the network, as the potential would now exist to mitigate the damage from compromised nodes permanently. Key management systems have been proposed for wireless sensor networks which aim to provide cryptographic protection of communication as well as sensor capture detection [16]. While these systems are useful, they do not identify when a node is captured and reprogrammed based on a change in its behaviour. Using ADIOS, keys may be dynamically revoked based on a node's behaviour as voted on by its neighbours.

D. Black Hole Attacks in Wireless Sensor Networks

Black-hole attacks, otherwise known as packet drop attacks or sinkhole attacks, are a type of denial-of-service attack that can be employed against wireless sensor networks [5]. A popular choice of routing protocol for use in wireless sensor networks and mobile ad-hoc networks is the Ad-hoc On-Demand Distance Vector (AODV) protocol. This is a reactive routing protocol in which nodes cooperate to find routes to the destination requested by a node originating a route request [17]. A black-hole node on a wireless sensor network running AODV may potentially advertise a short route to the sink node in an effort to attract more packets. This node would then drop the packets, never forwarding them on to the sink node [18]. Needless to say, this is a potentially devastating attack on a wireless sensor network that does not have any black-hole defence mechanism.

This paper proposes a system of implementing a lightweight expert system on each node in a wireless sensor network, to allow the nodes to make judgements on the behaviour of their neighbouring nodes. For example, it would make sense for a node, Node A, on the network to raise an alarm if it overheard another node, Node B, announcing a path to a particular destination (with Node B as one of the hops on the route) and then not hear Node B pass on the corresponding data it received towards the destination.

III. Using the CLIPS-based Expert System to Secure Against Black Hole Attacks

To detect black-hole attacks on a wireless sensor network, a node can simply listen for route announcements from neighbouring nodes, and then make a note of whether or not the announcing node actually forwards packets that come as a result of the aforementioned announcement. This is the basis of the watchdog and pathrater system [19], however ADIOS makes a distinction by using a CLIPS-based lightweight expert system to make decisions and interpretations of what the incoming data means.

The architecture for this system was developed with two main design guidelines in mind. The first is to develop a means of integrating expert systems with wireless sensor networks in a way that is robust and easily extensible. The second is to respect the physical limitations of nodes [4] by devising an intrusion detection and prevention system that will require only moderate amounts of processing power and memory to operate.

Fig. 1 illustrates the overall architecture of ADIOS, the expert system based intrusion detection and prevention system. As depicted by the diagram, the receiver is placed in promiscuous mode and this allows the mote to observe all transmissions within its range. ADIOS relies on a sensor node's ability to overhear the transmissions of its neighbouring nodes for it to make useful inferences about intrusions on the network.

A. How the Expert System Interfaces with Actual Network Events

The transmissions that are overheard are fed to the CLIPS backend which determines, based on the rules that were developed for the expert system, whether or not they are

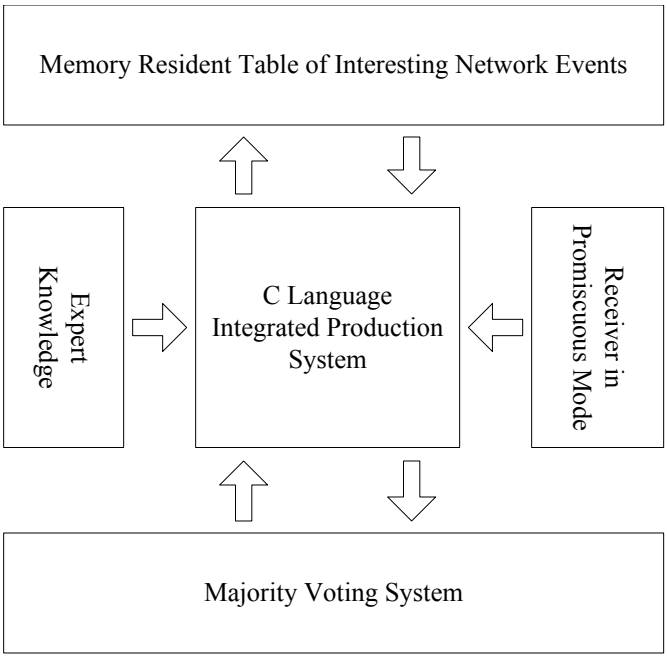

Fig. 1. Architectural diagram of the Intrusion Detection System that ADIOS uses.

significant, and how to handle them. Significant events are continually added to the memory resident table and this table forms the basis of the "knowledge" that the intrusion detection system possesses.

At preset intervals, the processing mechanism of our intrusion detection system is triggered and this causes the expert system to make inferences from the facts contained in the memory resident table. The CLIPS backend uses its rule-base to reason on the captured events and make determinations on whether or not suspicious activity is happening on the network. After each triggering of the intrusion detection process, suspicious events or neighbours, if any, are identified and the memory resident table will be empty or nearly empty.

The design of ADIOS is what makes the memory resident table empty or nearly empty at the end of each triggering of the intrusion detection process. Network events and transmissions are asserted as facts and these facts are what the expert system reasons on. Facts corresponding to expected events are eventually retracted since ADIOS would no longer need to be concerned with them. Facts corresponding to misbehaviour are analysed to obtain data about the misbehaviour as well as the node responsible for the misbehaviour. A single new fact containing the data of each misbehaviour is re-asserted, while the original facts that highlighted the misbehaviour are retracted. This is why the memory resident table will contain significantly less facts at the end of each intrusion detection run.

Depending on the severity of misbehaviour that is identified, the CLIPS backend may manipulate metrics in the local routing table or communicate with the majority voting system for further action to be taken against a misbehaving node.

1) Manipulating metrics in the local routing table: Suppose a neighbouring node is behaving erroneously (perhaps because it is overloaded and lacks CPU cycles) but this behaviour is deemed to be of low severity. It might be judicious to locally modify the metric of routes that have that neighbour as a next hop instead of invoking the majority voting and key revocation process and potentially expelling the neighbour from the network. By simply manipulating local routing metrics in response to such misbehaviour, a node may prefer a different route to the same destination without unduly burdening the wireless sensor network with majority voting.

2) Invoking the majority voting system: If, on the other hand, a node is outrightly participating in routing misbehaviour, ADIOS has the option of invoking a majority voting process. A potential outcome of this process is that the misbehaving node's neighbours will agree that it is behaving outside of design and it could then be evicted from the network using key revocation or some other mechanism.

It is important to note that the expert system chooses what to do after suspicious activity is detected based on rules in its rule-base. This is an area where customisations can be made, in that the nodes on the network can be made more or less "forgiving".

ADIOS is designed with modularity in mind and so each component may be upgraded independently of the others. This fits well with the first design guideline of having a modular intrusion detection and prevention system with no single point of failure. Once the CLIPS-based expert system is up and running on the nodes in a wireless sensor network, the particular rule-base that CLIPS uses may be upgraded, modified, or reused based on the requirements of the particular sensor network without necessitating a change in the sensor network architecture or the architecture of the intrusion detection and prevention system.

B. How CLIPS can be used make inferences from incoming data

By representing malicious node behaviour or attack signatures as rules in the expert system backend, ADIOS can use its inference engine to identify anomalies when fed with network data. Note that it may require multiple complex rules to properly describe certain denial-of-service attack signatures or node misbehaviour.

Fig. 2 shows a simple CLIPS rule that can be used to identify data modification. Suppose some data with checksum of 0x1532 was sent from Node A to the next hop of Node B and that this checksum was stored in ADIOS' memory resident table. Further suppose that when Node B passed on the data, Node A overheard the transmission and recorded a new checksum of 0x298F in ADIOS' memory resident table. The CLIPS rule in Fig. 2 would be able to identify the data modification anomaly when the intrusion detection process was invoked by comparing the expected checksum to the new checksum.

From the CLIPS rule in Fig. 2 we also see the design strategy of retracting the facts that indicate anomalies and replacing them with a single new fact that represents the anomaly. This helps to conserve memory and falls in line

```
(defrule DETECT_MODIFICATION " "

(sent (id ?a) (checksum ?b))
(heard (id ?a) (checksum ?c))
(test (not (= 0 (str-compare ?b ?c))))
?fact_a <- (sent (id ?a) (checksum ?b))
?fact_b <- (heard (id ?a) (checksum ?c))

=>

(retract ?fact_a)
(retract ?fact_b)
(assert (anomaly (name "MODIFICATION")
(id ?a) (checksum ?b) (checksum ?c)))

)
```

Fig. 2. Simplified example of a CLIPS rule that can be used to detect data modification.

with the second design guideline of making ADIOS resource-friendly enough to be used on low-power wireless sensor network nodes. The new facts representing anomalies are then handled either by majority voting or manipulating routing metrics and also retracted. This means that at the end of the intrusion detection and attack mitigation process, the memory resident table will be empty and waiting for a new round of network events.

C. Why the Proposed System is Robust and Extensible

Denial-of-service attacks and node misbehaviour are captured in the form of rules, which are written in a file and then loaded into CLIPS. This input file that CLIPS uses, hereafter referred to as the "expert-knowledge", is what tells the CLIPS-based expert system backend what series of events correspond to a potential attack. This expert-knowledge is used to analyse incoming network events by specifying the types of inferences and conclusions that should be made based on the particular observations that are fed into the system.

Expert-knowledge can continuously be updated and improved if and when new weaknesses are discovered and this allows for the existing intrusion detection and prevention framework to be used while offering the extensibility of using new or different expert-knowledge customised based on network type or other network parameters, for example. Obviously, an intrusion detection and prevention system running on a wireless sensor network monitoring soldiers behind enemy lines may be expected to behave differently than one which is monitoring the moisture content of soil in a backyard garden. ADIOS allows the operator of the sensor network the option to customize the expert-knowledge protecting their sensor network from a mode that gives battery life a priority to a mode that is designed to prevent denial-of-service attacks at any hardware resource cost.

D. Simulating the Expert-system Based Approach to Protecting Wireless Sensor Networks

In order to implement the lightweight expert system on motes, the CLIPS engine would be stripped down and loaded as a software module on the existing firmware. For the purpose

of testing this approach, the CLIPS Java Native Interface (JNI) has been utilised. The CLIPS JNI is a library that allows Java programs to communicate with a CLIPS backend. This interface allows one to write Java programs that are able to utilize CLIPS-based expert-knowledge at the backend and allows the convenience of simulating the performance of the proposed intrusion detection and prevention system by using small Java programs behaving as if they were actual motes.

Expert-knowledge is loaded into these simulated motes and then tools such as ns-2 or ns-3 [20] may be used to introduce irregularities into the network. We can then monitor the responses of our simulated motes to determine if the attack was detected, how long it took to detect the attack, and how long it took to identify a misbehaving node on the network. Of course, the misbehaving node would be punished in some way, and this punishment could be determined using expert-knowledge customised for the particular scenario.

Also of interest are the memory and processing requirements for running lightweight expert systems on low-power motes. Several optimizations for making ADIOS lightweight and resource-friendly have been examined and are discussed in Section IV. The results of the tests conducted on our simulated "expert motes" will facilitate the determination of what an ideal lightweight expert system for low-power wireless sensor network nodes would look like as well as the particular approaches to engineering expert-knowledge to make the system perform better in practice.

IV. Monitoring Network Events using an Expert System in a Resource-friendly Manner

The low-cost, low-power nature of wireless sensor nodes reduces the amount of headroom available in terms of computations that we can do or are willing to do on them. CLIPS is a small, lightweight, portable, open-source expert system tool designed to be easily integrated into existing systems [8]. This makes CLIPS a solid base to build a lightweight expert system, such as the one presented here, for use on low-power motes. Although CLIPS was utilised for this particular system, it is conceivable that any similar lightweight expert system building tool could be used.

Considerations such as number of CPU cycles required and amount of memory required are just a few of the many considerations to be made when deploying any system into an environment. There were initial concerns about the amount of resources that would be required to successfully run an expert system on a low-power and/or low-memory device. Below, we suggest possible optimizations for a resource-friendly deployment.

A. Optimizations for Memory-constrained Systems

In devices with severe memory constraints, the amount of interesting network events that is fed into the expert system can be minimised so as not to overwhelm the device. For example, a subset of interesting network events or packets may be monitored in an effort to keep memory consumption low. By randomly choosing what packets to monitor, minimal resilience to attack is sacrificed, as an adversary attempting to trick the system by selectively forwarding packets could not know beforehand which packets are being monitored.

At the expense of more CPU cycles, we can process our network events more often to save memory. Recall from above that the memory resident table is emptied each time the processing mechanism of our intrusion detection system is run. By increasing the rate at which network events are processed, the peak memory consumption of the intrusion detection system can be reduced.

The expert-knowledge may also be designed to favour retracting facts when they are no longer useful before going on to processing other events. With properly engineered expert-knowledge, the expert system based method of intrusion detection and prevention can be ported to be compatible with a wide variety of low-memory devices, while still maintaining the capacity to detect and defend against denial-of-service attacks on the wireless sensor network.

B. Optimizations for Processing-constrained Systems

Optimizations for reducing the amount of CPU cycles required to run ADIOS include limiting the size of the expert-knowledge that is placed on each node based on the nature of the deployment. Each denial-of-service attack has a particular signature which will be represented as rules in the rule-base. Describing an attack signature using the CLIPS expert system backend may require anywhere from a few simple rules to many complex rules. Thus an intrusion detection system capable of identifying a wide range of attacks will have a larger rule-base and consequently require more CPU cycles on each run. More powerful nodes used in military applications may want to favour analysing network events for all known attacks, while less powerful nodes for civilian use may sacrifice some resilience to attack in favour of less CPU usage and longer battery life.

Monitoring a subset of interesting network events or packets as detailed above also helps to reduce the computation required since the memory resident table would be smaller and thus we can invoke the intrusion detection mechanism less frequently. The power of the intrusion detection framework presented here comes from giving the user the ability to defend their network based on their requirements and the capabilities of the hardware that they have.

V. Conclusion and Future Work

Wireless sensor networks have shown tremendous utility and are beginning to see widespread deployment in many areas of everyday life. Because wireless sensor networking is a fairly recent innovation and wireless sensor nodes most times have significant resource constraints, the technology is not as mature in terms of security as other pervasive technologies. As long as we continue to become more reliant on wireless sensor networks, there is a need to ensure that confidentiality, integrity, and availability are maintained. ADIOS provides a means of enabling wireless sensor networks to defend themselves autonomously when attacked by an adversary. Preliminary simulations have suggested that lightweight expert systems may be used to mitigate denial-of-service attacks in wireless sensor networks.

We intend to extend our system and its corresponding CLIPS definitions to be able to detect and defend against a wide range of network security threats against wireless

sensor networks. Our research is currently limited to detecting and defending against black-hole attacks and certain limited types of node misbehaviour such as data modification. We envision a future version of ADIOS capable of mitigating threats to wireless sensor networks outside of denial-of-service threats. We intend to implement our lightweight expert system on sensor nodes in a medium-sized wireless sensor network deployment and do further experimentation to determine any other practical considerations to make when deploying expert system based intrusion detection and prevention systems in wireless sensor networks.

This research makes contributions to the domain of wireless sensor networking, network engineering, network security, expert systems and artificial intelligence. ADIOS is a means of enabling wireless sensor networks to be able to defend themselves autonomously when attacked by an adversary. This has the effect of making the technology more mature and reliable and, as a consequence, will foster the further adoption and implementation of wireless sensor networking.

ADIOS' novel approach to securing wireless sensor networks is expected to give adversaries far more work if they decide to attack a wireless sensor network since they would be attacking a dynamic, adaptive system and is expected to take us a step closer to ultimate security for the wireless sensor network.

ACKNOWLEDGMENT

The authors would like to thank Dr. Daniel Coore and Dr. Gunjan Mansingh for their invaluable contribution towards overall system design. We are also grateful to Dr. Busby-Earle for providing guidelines from a network security perspective. Garth Thomas provided technical support and assistance in preparation for deployment. Finally, we wish to thank the University of the West Indies for their gracious hospitality and willingness to provide outdoor space for testing purposes.

REFERENCES

[1] I. Akyildiz *et al.*, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] A. Perrig *et al.*, "SPINS: Security protocols for sensor networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 521–534, Sep. 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1016598314198

[3] ——, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004. [Online]. Available: http://doi.acm.org/10.1145/990680.990707

[4] J. Yin and S. Madria, "A hierarchical secure routing protocol against black hole attacks in sensor networks," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006.*, vol. 1, Jun. 2006.

[5] A. Wood and J. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, Oct. 2002.

[6] C. Hartung *et al.*, "Node compromise in sensor networks: The need for secure systems," Tech. Rep., 2005.

[7] H. Chan and A. Perrig, "Security and privacy in sensor networks," *Computer*, vol. 36, no. 10, pp. 103–105, Oct. 2003.

[8] G. Riley, "Clips: An expert system building tool," in *The Second National Technology Transfer Conference and Exposition, NASA, Washington, Technology, 2001*, vol. 2, 1991.

[9] C. Kaufman *et al.*, *Network security: private communication in a public world, second edition*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2002.

[10] D. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[11] S. Russell *et al.*, *Artificial intelligence: a modern approach*. Prentice Hall, Englewood Cliffs, NJ, 1995, vol. 2.

[12] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 275–283. [Online]. Available: http://doi.acm.org/10.1145/345910.345958

[13] J. Deng *et al.*, "A performance evaluation of intrusion-tolerant routing in wireless sensor networks," in *Information Processing in Sensor Networks*, ser. Lecture Notes in Computer Science, F. Zhao and L. Guibas, Eds. Springer Berlin Heidelberg, 2003, vol. 2634, pp. 349–364. [Online]. Available: http://dx.doi.org/10.1007/3-540-36978-3_23

[14] F. Chou and J. Tan, "A majority voting scheme in wireless sensor networks for detecting suspicious node," in *Second International Symposium on Electronic Commerce and Security, 2009. ISECS '09.*, vol. 2, May 2009, pp. 495–498.

[15] G. Dini and I. Savino, "An efficient key revocation protocol for wireless sensor networks," in *International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2006*, 2006, pp. 450–452.

[16] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM conference on Computer and communications security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 41–47. [Online]. Available: http://doi.acm.org/10.1145/586110.586117

[17] I. Chakeres and E. Belding-Royer, "AODV routing protocol implementation design," in *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.*, Mar. 2004, pp. 698–703.

[18] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.*, May 2003, pp. 113–127.

[19] S. Marti *et al.*, "Mitigating routing misbehavior in mobile ad hoc networks," in *International Conference on Mobile Computing and Networking*. ACM, 2000, pp. 255–265.

[20] E. Weingartner *et al.*, "A performance comparison of recent network simulators," in *IEEE International Conference on Communications, 2009. ICC '09.*, Jun. 2009.