Formal modelling and analysis of routing protocol security in wireless sensor networks

Kashif Saghar*, William Henderson and David Kendall
School of Computing, Engineering and Information Sciences
Northumbria University, Newcastle upon Tyne, UK
*Email: Kashif.Saghar@unn.ac.uk

***Abstract*—Wireless Sensor Networks (WSN) are composed of small, low cost, resource-constrained computing nodes equipped with low power wireless transceivers. Generally, they are embedded in their environment to perform some specific monitoring and/or control function. Unlike wired networks that have dedicated routers for network connectivity and message forwarding, every node in a WSN can act as a router in a multi-hop network. A WSN can offer a cheap, application-specific solution in a variety of situations including military and disaster response scenarios, where other approaches are not viable. Due to their unattended nature and deployment in possibly hostile environmental conditions, there are many challenges in ensuring that a WSN is formed effectively and survives long enough to fulfil its function. Securing a WSN against attack is a particular challenge. Traditional encryption mechanisms are resource hungry and are not sufficient alone to provide a complete solution. This project is concerned with secure routing protocols. Formal methods are used to model and analyse the design of existing protocols and to demonstrate some previously unreported weaknesses.**

***Index Terms*—Wireless Sensor Networks (WSN), Routing Protocol, Formal Modelling, DoS (Denial of Service) Attacks, Protocol Verification.**

I. Introduction

Wireless Sensor Networks (WSN) are composed of low cost, low power, small computing nodes, communicating by wireless to monitor and/or control some aspect of the environment in which they are embedded. The number of nodes in a WSN may vary from a few to a few thousand. Unlike wired networks, that have dedicated routers for network connectivity and message forwarding, every node in a WSN may be called upon to act as a router. The main aim is to route data from nodes that detect some event in the environment (source) to nodes that require information about that event (sink or base station). Routing is the process of moving data between nodes from source to sink. A routing protocol determines which path(s) the data should follow. Due to their broadcast transmission, limited resources, unattended nature and hostile environmental conditions, there are many challenges to ensuring the secure and reliable operation of a WSN. This work addresses the problem of the vulnerability of WSNs to denial of service (DoS) attacks leading to the disruption of the flow of data from source to sink.

Many DoS attacks have been recognised in the literature [11], [15]. The vulnerability of routing protocols to these attacks has been discussed and new protocols have been developed, or older protocols modified, to guard against them [11].

But the techniques used so far generally rely on visual inspection and simulation which are often not adequate for the detection of worst case scenarios. It is possible that bugs and vulnerabilities remain. The work described here investigates to what extent the application of formal methods leads to more effective bug detection in routing protocols for WSNs. By using formal methods, not only are the assumptions of protocols clearly modelled, but also all possible behaviours of the models can be examined for error conditions.

Our aim in this research is to develop a formal framework that can be used to check the resilience of WSN routing protocols to denial of service attacks. Finite state models of protocols are described using PROMELA [7] and simple liveness properties are checked using the SPIN model checker [7]. Properties are checked for all possible topologies of N nodes, where N is typically small (< 10) in order allow the automatic verification to complete with the computing resources (memory and CPU time) available. This has proved to be an effective bug-hunting approach and weaknesses in existing protocols have been discovered.

The rest of this paper is organised as follows: section II briefly discusses related work; denial of service attacks are described in section III; our formal framework is introduced in section IV and its application to existing protocols is described in section V. Conclusions and further work are presented in section VI.

II. Related Work

Other researchers have realized that computer simulation is often inadequate for finding errors in routing protocols and the development of formal models to check various aspects of routing protocols is becoming more common. Formal models have been used also in the analysis of security protocols. A representative selection of this work is identified below.

TinySec and LEAP have been modelled using the high-level formal language HLPSL [14]. The authors found two attacks, a man-in-the-middle attack and a type flaw attack, which show that confidentiality is compromised and an intruder may obtain confidential data from a node in the network. They also checked SNEP [13], which is the base component of the security protocol 'Security Protocols for Sensor Networks' (SPINS) and by formal analysis disclosed an attack that

allowed the acceptance of a false request message from an intruder. Y. Hanna et al. [5] developed a new approach (Slede) that extracts a PROMELA model of a protocol from its implementation in NesC, a topology description and an intruder template library. They verified μTESLA [1] and LEAP [12] using this method.

Other work has used SPIN to analyse security properties in extensions of ad hoc routing protocols based on Dynamic Source Routing (Ardiadne and endairA) [2]. The authors have used an automated security evaluation process and analysed all topologies for networks of up to 5 nodes. Our work extends the application of this approach to routing protocols for WSN.

III. Denial of Service Attacks

We use the term *denial of service* in a general sense to mean the adverse effect of any malicious external agent (attacker) on the correct or timely delivery of data from source nodes to sink nodes. The particular focus is on the effects of denial of service on routing protocols in WSNs. A brief summary of attacks that we have considered is presented below. More detailed surveys of attacks are available [11], [15].

Spoofing is the altering or replaying of routing information (data, beacon or acknowledgement packets). It can lead to the creation of inaccurate or unstable routes. *False injection* is the introduction of extra data or control packets into the network. It consumes bandwidth and may cause routing loops. The main aim of this attack is to consume resources wastefully. A *Sybil* attack occurs when a malicious node presents multiple identities simultaneously within the network. Such a node may be used to subvert routing protocols that rely on redundancy, such as multi-path protocols.

In a *black hole* attack, a malicious node joins the network and then either discards all the messages it receives or performs selective forwarding. A *sink hole* is a potent form of black hole in which the attacker makes itself particularly attractive to all nodes within its range, usually by advertising low cost routes to all destinations. In a *wormhole* attack, an attacker records packets (or bits) at one location in the network, tunnels them to another location, and retransmits them there into the network. A wormhole not only disturbs correct routing but is also the precursor to many other attacks black hole, sink hole, etc.

The *HELLO flood* attack involves the use of a high power transmitter by an attacker to broadcast routing or other information, with the purpose of convincing every node within radio range that the attacker is a neighbour. The attacker may then be established in routes that are unusable by other nodes since their transmitters are much less powerful.

Jamming is a physical layer attack instigated by creating radio noise in a particular physical area. *Traffic analysis* attacks are launched by capturing packets in order to estimate the location of the sending node. Thus, important nodes such as the base station can be identified and and targeted.

IV. Formal framework in SPIN

In order to rigorously check routing protocols against attacks we have developed a modelling framework that allows automatic analysis of protocols using SPIN. The main aim of the modelling is to enable checks against certain attacks for all possible network topologies. A network topology is represented by a boolean $N \times N$ matrix, where N is the total number of nodes in the network. A '1' in the matrix at (i, j) indicates a connection between nodes i and j; '0' indicates no connection. We assume symmetric or bidirectional links, so the value of (i, j) is equal to (j, i) for all i and j. Also, we assume that the channel is error-free, i.e. no message is lost in communication. The presence of bugs detected in such an ideal radio environment is a clear indication that a protocol will exhibit similar problems in a more realistic environment.

Primitive communication channels in PROMELA provide point-to-point communication. In order to represent the broadcast nature of the wireless medium, we model it as a process that enables communication between all nodes in radio range (as determined by the connection matrix). A more efficient analysis could be performed if PROMELA offered broadcast communication as a primitive, since the number of states in the model would be smaller. We apply some ad-hoc techniques to reduce the number of states to be explored. For example, in a unicast transmission only the destination node is interested in receiving data, so one can reduce the number of states to be considered by allowing other nodes to ignore such messages. There are other cases in which some nodes are not interested in broadcast messages, e.g. the base station does not need to receive its own transmissions. We have eliminated such messages to further reduce the state space. On the other hand, an attacker is normally eavesdropping on the channel, so we ensure that attackers are able to hear all messages transmitted within their range.

Typically, a model comprises a process for the wireless medium, as described above, and processes representing wireless nodes acting in one of the following roles: source, sink (base station), relay or attacker. Each wireless node process is instantiated from a PROMELA template that represents node behaviour as determined by the role in the protocol and attack under consideration. This approach is illustrated in Section V.

Given such a model, the behaviour of the network can be explored by using SPIN to check simple properties, expressed in linear-time temporal logic. Most often, the property of interest is a response property of the form 'whenever the source transmits a message, it is received eventually by the sink'. This property is checked both in the presence and absence of attackers. A great advantage of using a model-checker such as SPIN to test such properties is that a counter-example, in the form of a system trace, is provided automatically by the tool when the property fails to hold. The counter-example can be interpreted to determine the exact nature of the flaw in the protocol.

We have tested various published protocols to confirm the utility of our framework. The protocols that have been tested so far are TinyOS [11], Authentic TinyOS using μTESLA [1], Rumour Routing [3], LEACH [6], MCF [4], Directed Diffusion [8], Basic INSENS [10] and Enhanced INSENS [9]. The application of the framework to the first two of these protocols

is discussed in detail below.

V. Protocol analysis and results

A. TinyOS Protocol

We start our modelling with the TinyOS protocol [11] which is the simplest of all WSN routing protocols. This protocol constructs a spanning tree rooted at the base station. Data then flows from source nodes back to the base station using paths in the spanning tree.

Two types of message are involved: hello *beacon* messages and *data* messages. The notation $N_i \rightarrow N_j : M$ is used to denote that node N_i transmits message M to node N_j. A broadcast transmission is indicated by the use of $*$ in place of N_j and indicates that message M is transmitted to all nodes within radio range of N_i. A brief description of the protocol follows.

The base station periodically broadcasts a hello beacon which is flooded throughout the network.

$$B \rightarrow * : (beacon, ID_B) \tag{1}$$

A node, on first hearing a hello beacon, makes the transmitting node its parent and ignores any future beacons. The node then rebroadcasts the beacon with its own ID:

$$Node_i \rightarrow * : (beacon, ID_i) \tag{2}$$

This process is repeated throughout the network until the spanning tree is established.

A source node, upon sensing a significant event in its environment, unicasts the data back to its parent. Data messages are unicast from node to parent node until the base station is reached.

$$Node_i \rightarrow Node_{parent} : (data, ID_{parent}, PAYLOAD) \tag{3}$$

A simple version of the protocol is modelled in which there is a base station B, source node S, relay N and attacker A. All *beacon* messages are broadcast; *data* messages are unicast. It is assumed that the attacker can hear all messages. For various attacks, we check the required property that data always reaches the base station from the source node, in all topologies where there is a path between them. The modelling of attacks is discussed below.

1) Spoofing: In a spoofing attack, the attacker transmits the hello beacon using the *ID* of the source node instead of its own *ID*. The required property is violated and the trace reveals that the problem is caused by the creation of a routing loop in topologies in which the source node and the attacker share a neighbour, since the source node and its neighbour are established as each other's parent and data loops between them forever. Fig. 1 illustrates one such topology, detected using SPIN from an analysis of all 4-node topologies.

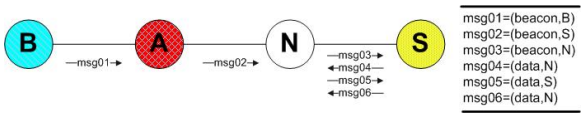

Fig. 1. Topology for which SPIN traces a successful spoofing attack

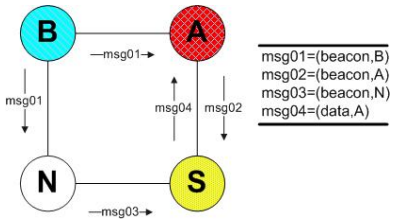

Fig. 2. Topology for which SPIN traces successful black hole and sink hole attacks

2) Black hole: A black hole is modelled simply by having the attacker forward *beacon* messages correctly but drop *data* messages randomly. The required property is violated and the trace confirms that when the attacker is a parent in the data forwarding path, data may not reach the base station. Fig. 2 illustrates a successful black hole attack in one topology detected using SPIN. Note that here node A behaves normally during beacon forwarding but drops data once it becomes the parent of the source node S.

3) Sink hole: We model a sink hole attack by having not only the base station, but also the attacker, initiate the broadcast of hello beacons. The attacker also ignores legitimate beacons from the base station. In this model, the required property is violated and the trace confirms that when the attacker acts as a sink hole, the source and intermediate nodes transfer data to it rather than the base station. Fig. 2 illustrates one such trace detected using SPIN in an analysis of all 4-node topologies. In this example, node A acts as a base station and broadcasts a hello beacon (msg01). The source node receives this beacon before the legitimate beacon (msg03) and so transfers data to the attacker (msg04).

Note that for this simple protocol in such a small network, the scenarios illustrating the sink hole and black hole attacks are identical. However, the behaviour of the attackers is different in each case and, in general, the scenarios illustrating the attacks will be different. This is shown clearly when considering the resistance of the Authentic TinyOS protocol to each type of attack.

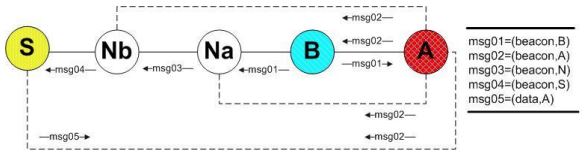

Fig. 3. Topology for which SPIN traces a successful hello flood attack

4) HELLO flood: The high power transmitter of the attacker in a hello flood attack is modelled by modifying the connection matrix to represent the fact that the attacker can establish a unidirectional link with all other nodes in the network. Note that this is contrary to our usual assumption of symmetric links but is an accurate reflection of reality in this case.

Analysis of such a model shows that the required property is not satisfied and the trace reveals the cause of the problem. When the attacker broadcasts hello beacons it is heard by all other nodes which then establish the attacker as their parent in the spanning tree. Data from any source node whose transmitter is not powerful enough to reach the attacker is lost. Fig. 3 illustrates one 5-node topology, detected using SPIN, that illustrates the problem. Notice that msg02 is received by all nodes, including the source node S which ignores the legitimate beacon from node N and establishes A as its parent. The transmission of msg05 does not reach the attacker due to the low power transmitter of the source node. Of course, it is trivial for the attacker in this case to act also as a sink hole if it chooses.

5) Wormhole: In order to model the effects of a wormhole, the channel process can be modified to create a tunnel between the base station and a source node. The attacking node is hidden and all other nodes behave in the normal way. The tunnel only forwards beacons from the base station to the source node. Note that the attacker does not transfer data received from the tunnel, as the aim of a wormhole attack is to drop data transmitted in the tunnel.

Analysis of this model shows that the required property is not satisfied and the trace reveals that when a wormhole exists between the base station and a source node, the source establishes the base station as its parent instead of one of the nodes within its range; thus, when it transmits data, the data is lost as it is addressed to a node that is reachable only via the wormhole. Fig. 4 illustrates a successful wormhole attack in one of the topologies detected using SPIN in an analysis of all 4-node topologies.

An alternative approach to the modelling of a wormhole can be adopted, in which the creation of the wormhole between two distinct attacking nodes is represented explicitly. The

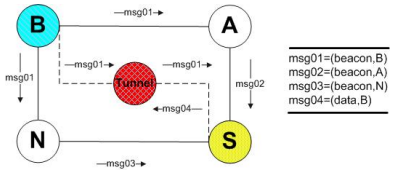

Fig. 4. Topology for which SPIN traces a successful wormhole attack

disadvantage of such an approach is that it increases the size of the state space that needs to be explored in the analysis. Our model is simpler and simulates the effect of the tunnel rather than its creation. In circumstances where it is the creation of the tunnel that is the primary object of study, the more costly modelling approach should be adopted.

6) Other attacks: We have also studied the effects of the false injection and Sybil attacks on the TinyOS protocol. In fact, the sink hole attack discussed earlier relies on the false injection of a hello beacon by the attacker. The Sybil attack has been modelled by having the attacker transmit hello beacons using both its real *ID* and a fake *ID*. The required property is not satisfied in this case and the trace shows that neighbours of the attacker might establish a non-existent node as their parent and then attempt to forward data to it. In the TinyOS protocol this attack resembles a spoofing attack.

B. Authentic TinyOS using μTESLA

The Authentic TinyOS protocol uses μTESLA [1] to authenticate messages. μTESLA is a lightweight authentication protocol aimed at reducing the computing resource requirements of the nodes that implement it. This makes it practicable for use with the resource constrained nodes of a WSN. The Authentic TinyOS protocol is similar to the TinyOS Beaconing protocol considered in Section V-A. The main difference is that the base station uses a μTESLA key, K_i, in the ith beaconing interval to generate a message authentication code (MAC) for the beacon. K_i is derived using a public hash chain whose seed is known only to the base station and thus no other node can generate or predict the next key. The base station starts by broadcasting the following message with t=0:

$$B \rightarrow * : (beacon, ID_B, MAC_{K_{t+1}}) \tag{4}$$

The receiving node checks if the beacon has been received in the first beaconing interval i.e. that

$$(T_{cur} + \delta - T_o)/T_{int} < I_i + d,$$

where T_{cur} is the current time at the receiver, T_o is the start time of the first beacon interval, T_{int} is the interval between beacons, δ is the maximum synchronisation error between the base station and other nodes, d is the key disclosure delay in seconds, and I_i is the interval number (i.e. 1,2,3,..). If this does not hold, the beacon is ignored otherwise it is pushed into a

FIFO queue as the pair (*ID*, *MAC*) and the node broadcasts the beacon, using its own *ID*.

$$Node_i \rightarrow * : (beacon, ID_i, MAC_{K_{t+1}}) \quad (5)$$

This process is repeated throughout the network. When the next time interval begins, the base station discloses the key for the previous interval by broadcasting:

$$B \rightarrow * : (beacon, ID_B, K_{t+1}, MAC_{K_{t+2}}) \quad (6)$$

This is again broadcast to all neighbours and each node chooses the first authentic beacon sender as its parent and then ignores further beacons. Each node also retransmits the authentic beacon.

$$Node_i \rightarrow * : (beacon, ID_i, K_t, MAC_{K_{t+1}}) \quad (7)$$

It is not necessary to model the details of the authentication mechanism to undertake a useful formal analysis of the protocol. It is assumed that μTESLA provides a reliable authentication mechanism. The focus of the analysis is on the effect of attacks on the protocol, given this assumption.

In order to model the Authentic TinyOS protocol, the model of the TinyOS Beaconing protocol is modified so that the base station sends two beacons: the first with a null Key field and the second with a key that can be used easily to authenticate the previous beacon. The message format is now (Type, ID, Key, MAC). The MAC is modelled simply by using a number. The base station sends the first message as $(beacon, B, 0, 1)$ i.e. the MAC is 1 and the Key is null. Then the second message $(beacon, B, 1, 2)$ is sent, where the key 1 will authenticate the previous MAC and the new MAC is set to 2 for future use. Using this simple model, the whole protocol can be analysed. Nodes check the authenticity of a beacon by testing if the MAC field of an entry (ID, MAC) in the FIFO matches the Key received in the beacon. Beacon intervals are modelled simply by waiting until all nodes have received a beacon before a new beacon is initiated by the base station.

Analysis of this model shows that the Authentic TinyOS protocol successfully resists the sink hole and false injection attacks to which the simpler TinyOS Beaconing protocol was shown to be susceptible in Sections V-A3 and V-A6.

However, our analysis shows that several other attacks still succeed, despite the use of authenticated broadcasts. Fig. 5 shows a successful spoofing attack, Fig. 6 shows a successful black hole attack, Fig. 7 shows a successful hello flood attack, and Fig. 8 shows a successful wormhole attack. All of these scenarios were discovered using the counter-examples generated by SPIN in an analysis of all 4-node and 5-node topologies.

C. Other Routing Protocols

We have modelled and checked other routing protocols such as Minimum Cost Forwarding (MCF) [4], Rumour Routing [3], LEACH [6], and Directed Diffusion [8]. Our approach confirmed that they are susceptible to spoofing, black hole,

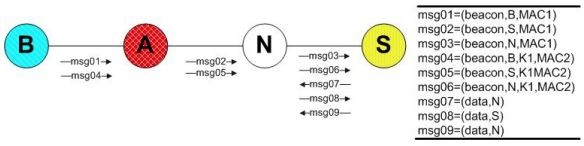

Fig. 5. Topology for which SPIN traces a successful spoofing attack

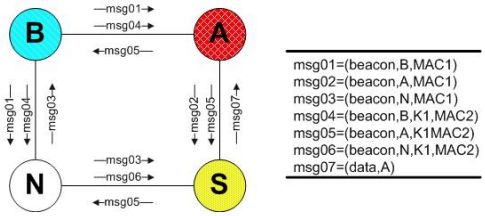

Fig. 6. Topology for which SPIN traces a successful black hole attack

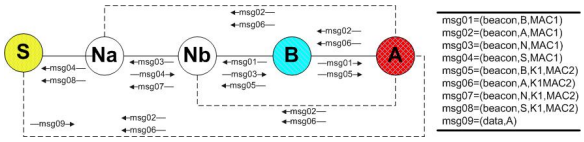

Fig. 7. Topology for which SPIN traces a successful hello flood attack

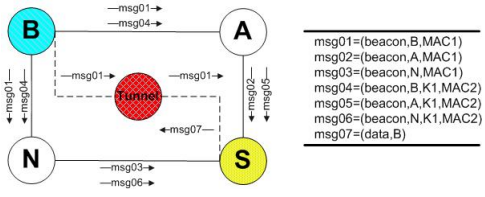

Fig. 8. Topology for which SPIN traces a successful wormhole attack

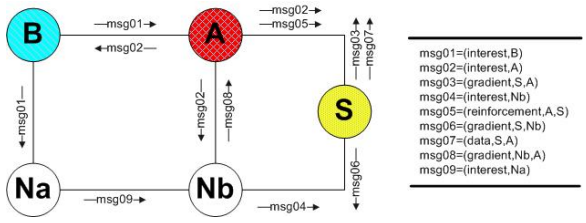

Fig. 9. Topology for which SPIN traces a successful sink hole attack against the Directed Diffusion protocol

sink hole, hello flood, wormhole and Sybil attacks. These results are as expected; these protocols have already been identified as susceptible to such attacks in several literature reviews. But we have also obtained some interesting results using our approach which have not been mentioned before. These are discussed briefly below.

First of all let us consider Fig. 9, showing a model of the Directed Diffusion protocol in the presence of a sink hole attack. Here the attacker behaves just like the base station. As soon as it hears an interest message from the legitimate base station, the attacker replays that message, identifying itself as the base station. Thus the source node will now send data matching the interest message to both the attacker and the legitimate base station. We model the effect of a sink hole attack by flooding an interest message from the attacker in which the ID is shown as the ID of the base station. Earlier research has suggested that data in this case will reach both the attacker and the base station. But our SPIN trace in Fig. 9 confirms a worse scenario in which sometimes data may never reach the base station in the presence of a sink hole attack. From Fig. 9, it is seen that the attacker floods an interest message (msg02) as soon as it receives a legitimate one (msg01) from the base station. In this case, node *Nb* receives the interest message from the attacker (msg02) before the legitimate message from node *Na* (msg09), which it then ignores. Thus, the gradient is set only towards the attacker, node A. In this way, a gradient is never established with node *Na* and so data will never reach the base station. Note that whenever the base station broadcasts its interest message again, the same process can be repeated.

Secondly, we have discovered an unreported bug in the Rumour Routing protocol [3] using our model-checking approach. The bug can be illustrated using the same topology as for Directed Diffusion, shown in Fig. 9. Consider the case in which a query has visited all nodes before the agent has started. In that case, the agent arrives at each node after the query has been passed on. The exhaustive search made by SPIN reveals the problem. Suppose in the topology in Fig. 9 the query has taken a path of $B - Na - Nb - A - B - A$. Note, as node A has no unseen neighbours, it forwards the query again to node B and node B forwards it back to A since it sent it last time to node *Na*. Now suppose the agent is started and takes the path $S - Nb - Na - B$. In this case, the query and the agent will never meet and the required property is violated.

Note that it is extremely unlikely that this scenario will be detected using visual inspection even in small networks, but that formal analysis of all possible topologies of a 5-node network reveals it quickly.

VI. Conclusions and Future Work

In this paper, we have considered the modelling and analysis of a simple routing protocol (TinyOS Beaconing) using SPIN. Other researchers have already described its vulnerability to attack and we have confirmed these results using our formal framework. We have checked a simple requirement property that whenever a source node transmits a data message it is received eventually by the base station. The requirement property has been checked both in the presence and absence of various attacks. Analysis shows that the property may not be satisfied in the presence of any of the attacks that have been considered. The counter-examples generated by SPIN are extremely useful in determining the reason for the failure of the required property in each case.

We have applied the same approach to an analysis of a more robust protocol (Authentic TinyOS with μTESLA) and have confirmed that it can resist sink hole and false injection attacks but remains susceptible to other attacks.

Furthermore, the formal framework that we have adopted has revealed flaws in the Rumour Routing and Directed Diffusion protocols which, so far as we know, have not been identified before.

These results encourage us to believe that formal analysis by model-checking can be used successfully to discover flaws in WSN routing protocols that may not be discovered by visual inspection or simulation.

We intend to apply this approach to the analysis of more robust routing protocols and to use it to assist in the development of a new protocol that will be more resistant to denial of service attacks.

REFERENCES

[1] V. Wen D. Culler A. Perrig, R. Szewczyk and J.D. Tygar. Spins: Security protocols for sensor networks. In *ACM Mobile Computing and Networking*, volume 8, pages 521–534, Sep 2002.

[2] Todd R. Andel and Alec Yasinsac. Automated evaluation of secure route discovery in MANET protocols. In Klaus Havelund, Rupak Majumdar, and Jens Palsberg, editors, *Proceedings of 15th International SPIN Workshop on Model Checking Software (SPIN 2008),Los Angeles, CA, USA*, volume 5156 of *Lecture Notes in Computer Science*, pages 26–41. Springer, August 10-12, 2008.

[3] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA*, pages 22–31, October 2002.

[4] Songwu Liu Lixia Zhang Fan Ye, A. Chen. A scalable solution to minimum cost forwarding in large sensor networks. In *Proceedings of the tenth International Conference on Computer Communications and Networks (ICCCN), Scottsdale, AZ, USA*, pages 304–309, 15-17 Oct. 2001.

[5] Youssef Hanna, Hridesh Rajan, and Wensheng Zhang. Slede: a domain-specific verification framework for sensor network security protocol implementations. In *Proceedings of the first ACM conference on Wireless network security (WISEC 2008), Alexandria, VA, USA*, pages 109–118, 2008.

[6] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS'00)*, page 223, Jan 2000.

[7] Gerard J. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison Wesley, 2004.

[8] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.

[9] R. Han J. Deng and S. Mishra. Insens: Intrusion-tolerant routing for wireless sensor networks. In *Elsevier Journal on Computer Communications, Special Issue on Dependable Wireless Sensor Networks*, volume 29, pages 216–230, 2005.

[10] S. Mishra J. Deng, R. Han. The performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *IEEE 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03), Palo Alto, CA, USA*, pages 349–364, April 2003.

[11] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, volume 1, pages 293–315, September 2003.

[12] S. Setia S. Zhu and S. Jajodia. Leap: Efficient security mechanisms for large-scale distributed sensor networks. In *ACM Conference on Computer and Communications Security (CCS'03)*, pages 62–72, October 2003.

[13] Llanos Tobarra, Diego Cazorla, and Fernando Cuartero. Formal Analysis of Sensor Network Encryption Protocol (SNEP). In *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS 2007), Piscataway, NJ, USA*, pages 767–772, Pisa (Italy), October 2007.

[14] Llanos Tobarra, Diego Cazorla, Fernando Cuartero, Gregorio Daz, and Emilia Cambronero. Model Checking Wireless Sensor Network Security Protocols: TinySec + LEAP. In *Proc. of the First IFIP International Conference on Wireless Sensor and Actor Networks (WSAN'07)*, pages 95–106, Albacete (Spain), September, 2007. IFIP Main Series, Springer.

[15] A.D. Wood and J.A. Stankovic. Denial of service in sensor networks. In *IEEE Computer*, volume 35, pages 54–62, Sep 2002.