Cluster Computing (2019) 22:S14551–S14557
https://doi.org/10.1007/s10586-018-2352-3

Improving Adhoc wireless sensor networks security using distributed automaton

S. Venkatraman[1] · P. Arun Raj Kumar[1]

Received: 29 December 2017 / Revised: 22 February 2018 / Accepted: 1 March 2018 / Published online: 26 March 2018

Abstract
Wireless Sensor Network due to their Adhoc nature have emerged as a popular technology driving the growth of Internet of Things and Cyber Physical Systems. Wireless Sensor Networks due to their self-configuring nature are more vulnerable to attacks at all layers of the OSI model. Malicious attacks include packet drops, change in the data structure and mimicking other devices to destabilize the network. This work proposed an Intelligent Control Mechanism to secure WSN based on the principles of Distributed Automaton. Simulations carried out show the effectiveness of the proposed technique in not only improving the network performance but also reduction in time space complexities. The proposed techniques performs well even in heterogenous network by using the clustered behavior.

Keywords Cluster behavior · Control mechanism · Distributed automaton · Malicious behaviour detection · State space model

1 Introduction

In general, Adhoc Wireless Sensor Networks (AWSNs) are environment and application specific distributed networks. The aim is to monitor all the real-time activities and also the environmental behavioral elements that exist in several of the other situations.

It covers a wide geographical area with less installation and administration costs. With the rapid development of AWSNs, it plays vital role in remote monitoring applications such as, weather forecasting, disaster management, industrial automations, process monitoring, smart city organization systems, healthcare management systems, traffic control systems, asset tracking systems, military operations, wildlife monitoring systems, and etc. AWSN is vulnerable because of constraint in resources including memory, bandwidth, communication rage, energy, and computing power [1].

✉ S. Venkatraman
venkat.s87@yahoo.com; venkats23@gmail.com

P. Arun Raj Kumar
p.arunrajkumar@nitpy.ac.in

[1] National Institute of Technology Puducherry, Karaikal, Puducherry, India

Effective utilization of energy and reliable transportation of packets in ASWNs are key concerns to attain highly efficiency in monitoring and controlling systems [2].

The prime task of a sensor node is to gather data from neighboring within its radio range and communicate and update rest of the sensor nodes about their latest environmental information. If any sensor node in AWSN is compromised, it may leads to various attacks. Such as Sybil attacks [3], Distributed Denial of service (DDoS) attacks [4], flooding attacks [5], node replication attacks [6], wormhole attacks [7], selfish node attacks [8], and sinkhole attacks [9]. In a distributed network, it is challenging to detect intrusion or malicious nodes.

Although there are some enormous mechanisms of security attack detection that have been organized for the AWSNs, most such standing solutions can manage only a very limited number of such security attacks [10].

Key features which are affecting security and control mechanism in AWSNs are:

i) AWSNs capable to have huge heterogeneity of network elements.
ii) AWSNs communication media is non-isolated and vulnerable in nature.
iii) AWSNs having high mobility nodes and communication links are dynamic.

Rest of this paper is organized as follows: Sect. 2, discusses related work in control and monitoring operations of AWSNs. Section 3, defines background concepts. Section 4, discusses proposed Distributed Automaton System (DAS) architecture briefly. Section 5, focuses on dynamic descriptions to collect malicious behaviour information from the AWSNs. Finally in Sect. 6, we concludes the paper.

2 Related work

The actual development of the AWSN attacks has been based upon some or all of the key parameters that belong to the WSNs like the accuracy of attack detection, the scalability, resilience, self-configuration, privacy, interoperability and minimal overheads [11].

In general, AWSNs covers sensor nodes randomly, which have no prior information about the architecture of the networks. Sensor nodes location and arrangement of nodes in network changes regularly. For example, in traffic monitoring system, a vehicle can move its location according with driver's acceleration and time. This type of random coverage issues in AWSNs handled using geometric methods [12].

In [13], reliable and secure communications in WSNs is achieved through modeling networks by integrating Erdos–Renyi graph and random K-out graph with K-connectivity method. In [14], anomaly detection method in WSNs based on support vector machines (SVM) and sliding window protocols. It deployed in large scale AWSNs that achieves high accuracy in detection of black hole attacks and selective forwarding attacks without draining the sensor nodes energy. But spoofing and DoS attacks detection accuracies are significantly very low in SVM approach.

According to [15], detection of malicious behaviour in hostile environment is achieved through spontaneous watchdog nodes. Normally watchdog nodes monitor the neighboring nodes which comes under its radio region in AWSNs. This malicious behaviour detection approach is not suitable if the network is dynamic in nature.

Learning based intrusion detection method used in [16], to utilize correlation between identified finite number of features to capture various acceptable behaviour of the WSNs. Every feature represent unique parameter of the WSNs. For example, feature F_1 denotes spatial attribute of a node, feature F_2 denotes mobility of a node, and similarly feature F_3 denotes temporal attribute value. Based on instantaneous value of these features detection algorithm will classify legitimate and abnormal behaviour of AWSNs.

In a recent survey [17], identifies relatively very limited research work has been proposed to detect malicious behaviour in hostile environments. Article [18] forecast the growth of WSNs market investment from $0.45 billion in 2012 to $2 billion in 2022. Survey [19], discusses recent developments and potential synergies of AWSNs with various real world application domains. It addressed various open issues and challenges remains in integrated WSN environments.

Deployment of AWSNs in electrical power system environment and the enhancement of three subsystems in diverse smart gird applications such as power generation, delivery, and utilization is discussed in [20]. It represents that harsh environment conditions, variable link capacity and packet errors, reliability and latency are the performance metrics for AWSN enabled smart grid applications.

In [21], implementation of AWSNs based on open source hardware devices such as, Arduino and Raspberry Pi to monitor environmental related applications are discussed. The experimental results, reflects the enhanced utility of that new design.

In [22], presents the major needs and challenges of AWSNs in enhanced building automation systems. Article [23], reviews recent research proposals on smart agriculture and points out the need of AWSNs in agricultural services. Such as, green house, pest control, irrigation, fertilization, animal and postures monitoring, viticulture, and horticulture.

From the study of related work, we can predict AWSNs has produced high impact in real time applications. Due to remote accessing, remote administration, less maintenance costs, monitors large scale environments, supports heterogeneous technology and embedded with huge variety of real world domains. But AWSNs are vulnerable to various attacks and attack prevention and detection methods are incomprehensive, not efficient to detect new anomaly behaviors in the environment because of heterogeneity in technology. To address these challenges, we propose a distributed automaton based framework to protect and control AWSNs from various attacks.

3 Background

Formally a Distributed Automaton (DA) is a mathematical representation of low power constraint - computing device with 7-tuples.

$$DA = (S, \pi, C, \Delta, S0, I, L)$$

Where:

- S is a set of State where S_0 is a proper subset of S and is a set of initial State
- π is a set of actions (Classified as input, output or internal actions)

- C is a set of clocks
- Δ is proper subset of S x £(C) x π x S is the transition relation
- I is time invariant attribute.
- L is a location specificity attribute with prior propositions

£(C) represents real time clock values. If a node location is invariant with certain circumstances then clock time will start. In a DA transitions are allowed if it satisfies predefine time constraints (guard). For example, instantaneous description of a DA's is represented as follows: A structure of a DA instant state description is $[S_{i,j}, C_i]$ where $S_{i,j} \varepsilon S$ is a ith State of jth distributed automaton and $C_j \varepsilon V(X)$ is a real time clock X's value of jth distributed automaton.

A Network of DA (DAS) is the collaborative arrangement of DA_i, where $1 \leq i \leq M$, and $DA_i = (S_i, \pi_i, C_i, \Delta_i, S_{0,i}, I_i, L_i)$ for $1 \leq i \leq M$ (M is total number of DA in a DAS network). An instantaneous description of a DAS is an instant state description $(S_{i,j}, C_i)$ where $S_{i,j} \varepsilon S_1 \times S_2 \times S_3 \ldots \times S_M$ is a location vector where M is finite and $C_i \varepsilon V(X)$ is a real time clock X's value. For efficient state space representation we introduced clustering approach, each cluster consists of its own and unique real time clock 'r'. From this clustering approach we can define instantaneous description of a DAS become as $[S_{i,j}, r]$ where $S_{i,j} \varepsilon S_1 \times S_2 \times S_3 \ldots \times S_M$ is a position vector and $r \varepsilon R$ is an identification real time constraint of local cluster, the rest of this paper we will use this definition as DAS's instant configuration.

We prefer to use model checking tool to simulate our proposed distributed automaton architecture. Uppaal [24] is an integrated modelling tool, which is used to design, specify, validate and verify proposed network models of real time system.

4 Proposed system model

The Network of DAS models a collection of identical reactive entities (DA's) evolving concurrently and synchronously in response to a sequence of external stimuli generated by its environment and broadcast to all of them by means of a communication system. Concurrence is meant as a co-operation between DA's for the accomplishment of a given computational task, which is behavior of a DA, is affected by the DA's that are logically related to it. In general this behavior is also affected by the input data coming from the external environment.

DA's co-operation is based on the ability of each DA to observe other DA's. In the more general case, this visibility is global, that is information is broadcast to all DA's in the proposed system. The synchronizer unit provides the system timing, ensuring the simultaneous activation of all DA's, while control and computing activities of the entire system are spread over DA's.

The injector automaton represents the DA's in interface to the external environment. It distributes the information received from the environment to all DA's shown in Fig. 1. Data injection can be performed according to different modalities and protocols, depending on the specific application requirements.

5 Dynamic description

The behavior of the DAS in correspondence with a given sequence of external data consists of series of global states of the N/W system's situation. Each global state consist of every mobile agent's instantaneous state information. The sequence normally start with its initial global state, and each successive global states are attained from its antecedent by executing a transition step. This global state actually gathers the situations (State) of all DAs present in the system.

The functioning of the DAS is determined by the functioning of DAs. At each step all DAs undergo three phases:

(i) **Prelude:** each DA perceives the instance information of other DAs. In this is way every DA able to share information, and can be prepared in the prelude. The successive operation works on a dependable global state of the system state. As the state information is received through the communication system (the minimum requisite of which is the gossiping capability), it is stored in a local buffer.

(ii) **Behavior:** the current state, the observed state of other DAs, and the external datum (if any) trigger the state transition. As a result, the DA moves to new state. Values of data-items and affinities may be modified.

(iii) **Postlude:** the new state is assigned and made visible, that is, it is sent to the communication system.

The state transition sequences are loop-free and congregate to one of the possible acceptable states in a predefined finite number of times steps. Based on this we should formulate rules and state definition in order to avoid loops and also to contribute deterministic behavior. The general form of a state transition is **c (e)** → **a** where **e** is (are) the event(s) that prompts the transition, **c** is a condition that safeguards the transition from being taken unless it is true when **e** occurs, after successful transition **c (e)** system will perform an action **a.**

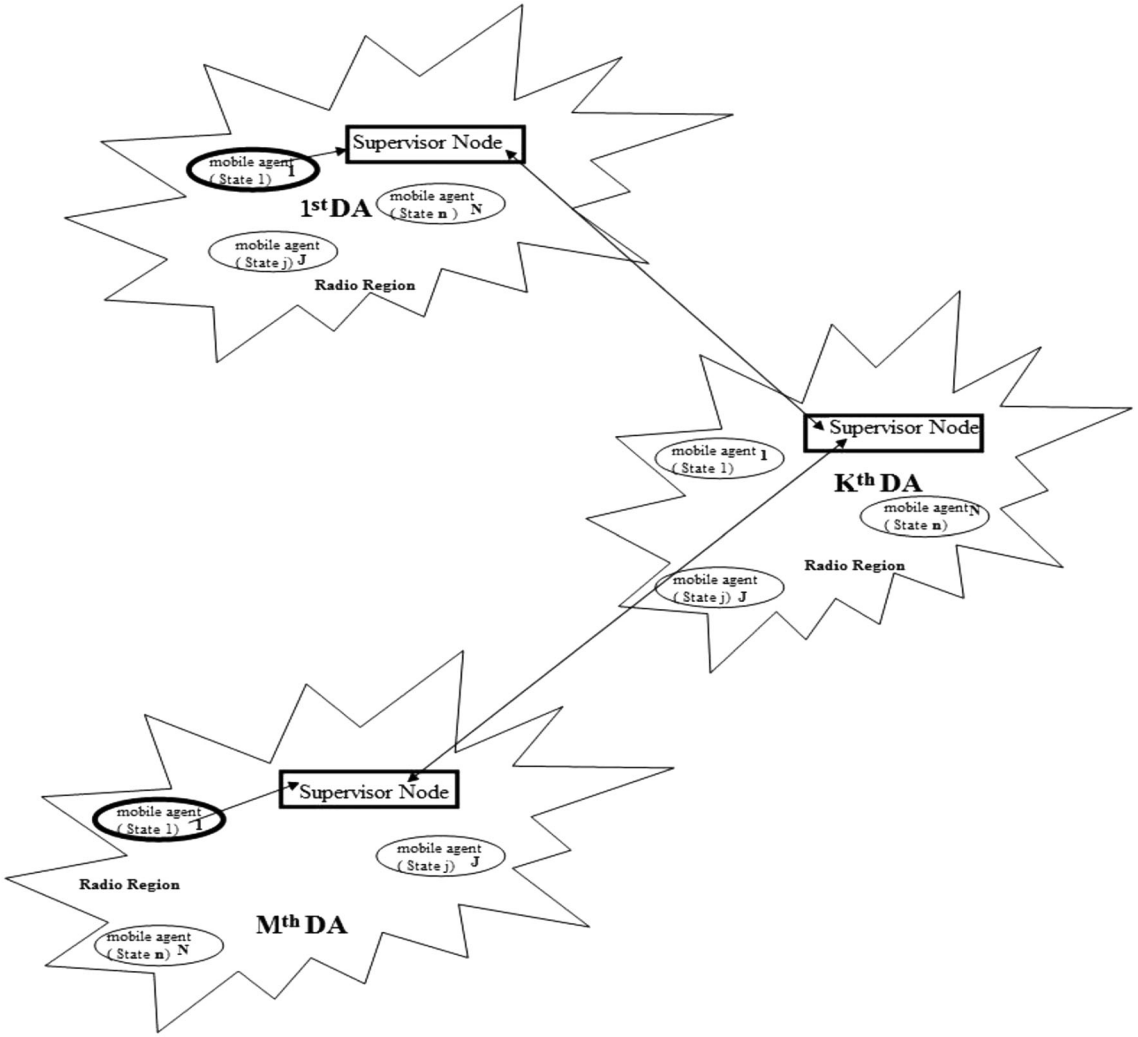

Fig. 1 Proposed system architecture of DAS

Fundamental for understanding the DAS is also the concept of stability. Indeed, it serves to characterize the correctness property of the system. Similarly with dynamic systems, the progress of the entire system can be viewed as passing from a highly unsteady state to more steady state, until a highly steady state, the acceptance state, is reached. This state might agree to an authorized termination of the computation (the highest steadiness) or to a malicious behavioral functioning condition.

Definition 1: A DAS is said to be in a stable state (or similarly to be stable), if and only if all DAs are in a stable state (or similarly are stable)

Definition 2: A DAS is said to be in a stable (or similarly to be stable), if there are no external stimuli from the environment and no transition rules can be applied

Definition 3: A DAS is said to be in an unstable state (or similarly to be unstable), if transitions occur from this state, that are unconditional to the external stimuli from the environment

Definition 4: A DAS is said to be in an unstable condition (or similarly to be unstable), if at least one DA is in an unstable state (or similarly is unstable) There are two kinds of stable conditions: those corresponding to final states and those corresponding to error conditions. Let us define the former as successfully stable conditions, while the latter as unsuccessful ones

Definition 5: A DAS is said to be in a successful condition (more briefly, it is successful) if all DAs are in a successfully stable state, or they are in the reset state

Definition 6: A DAS is said to be in an unsuccessful condition (more briefly, it is unsuccessful) if at least one DA is not in successfully stable state, nor it is in the reset state

Definition 7: A DAS system is said to be correct if and only if (1) for every legal sequence of external stimuli the system is successful and (2) for any illegal sequence of external stimuli some DA does not reach a successful stability condition nor it has left the reset state, that is the system is unsuccessful

The assumptions we have agreed in defining the DAS are the following:

1. A DA in an unstable state will always reach a stable state.
2. Changes occurring in a DA during a transition step sensed only after the end of the transition (in the prelude phase).
3. Changes occurring locally to a DA cannot be observed.
4. At each step DA's behavior is determined on the basis of the situation at the beginning of the step.
5. At each step, at most one transition per DA is taken.

5.1 Description of supervisor node

In this section we define Supervisor node as a distributed PDA and the detection of unexpected event in different modes.

Definition: An supervisor node is a 9—tuple DA $=$ $(Q, \pi, \Gamma, C, \Delta, q0, F, I, L)$ where,

1. Q is an n-tuple (Q1, Q2 … Qn) where each Qi are the set of states for Mobile agent i.
2. π is the finite set of legal actions.
3. Γ is an n-tuple (Γ_1, $\Gamma_2 \ldots \Gamma_n$) memory attribute.
4. C is a set of clocks
5. Δ is an n-tuple (Δ_1; $\Delta_2 \ldots \Delta_n$) of state transition functions. Where each

$$\Delta i : (Qi \times (\pi \cup \{\varepsilon\}) \times £(C) \times \Gamma_i) \rightarrow Qi, 1 \leq i \leq n$$

6. q0 ε U Qi is the initial state.
7. F ε U Qi is the set of final accepting states.
8. I is time invariant attribute.
9. L is a location specificity attribute with prior propositions.

Each of the component DA's of the Supervisor node is of the form

$$Ai = (S_i, \pi, C, \Delta_i, S_{0,i}, I, L); 1 \leq i \leq n.$$

Here S_i's needs not be disjoint. As in the case of DAS, we can have several modes of acceptance.

5.1.1 Broadcast-mode acceptance

Initially, the DA which has the starting state begins the processing of the input actions. For example the mobile agent i has the initial state. The processing precedes in mobile agent i as in a standalone DA. Assume in the mobile agent i the system arrives at a state q where q ε Qi. The ith mobile agent starts the processing to send percept message to supervisor node. The system goes to broadcast the message. The jth mobile agent ($1 \leq j \leq n$) provided q ε Qj. When j have a range of value from 1 to n, we can choose any one of them randomly. After choosing a certain jth component the DA remains in that cluster domain until it reaches a state outside the domain of its transition function and the above procedure is repeated. If there is an expected event then the automaton will reach any one of the accepting states. It does not matter which mobile agent the system is in or the stacks of the components are empty or not in supervisor node. The existence of stack rises the reproductive capacity of the complete system.

Definition: The instantaneous description (ID) of the supervisor node is given by a n + 3-tuple

(q, π, q_1, q_2,..q_n, C_i) where q ε Q, π ε set of actions, qi ε Qi, $1 \leq i, k \leq n$.

In this ID of the DAS, q denotes the current state of the whole system, π the portion of the input action yet to be read and i the index of the mobile agent in which the system is currently in and C_i clock time instant.

The transition between the dependable ID's is represented as follows:

$$(q, \pi_i, q_1, Xq_2 \ldots q_{n,} C_i) \vdash (q', \pi j, q_1, Yq_2 \ldots q_{n,} C_i)$$
$$\text{iff } \delta(q, \pi_{i,} X) \rightarrow (q', Y)$$

Where$(q, \pi, q_1, q_2 \ldots q_{n,} C_i) \vdash (q, \pi, q_1, q_2 \ldots q_{n,} C_j)$ iff q ε Q_j–Q_i and $1 \leq i, j \leq n$

Let $\vdash *$ be the transitive and reflexive closure of $\vdash$. Where X is percepts through sensor of specific mobile agent and Y represent the actions taken by the supervisor node in the DAS Model. An asymptotic analysis of time and state complexity of our proposed model compare with centralized nondeterministic finite state automata is given below in Table 1. Where m represents all possible states (instantaneous configurations) of an environment. Therefore number of states of the centralized finite control

Table 1 Comparison of computational time and state space size with centralized automaton

Model	Ability to represent maximum possible states	Computational time efficiency
NFA	N = O(m)	O(Nm)
DAS	S = O(m/H)	O(m/H + h)

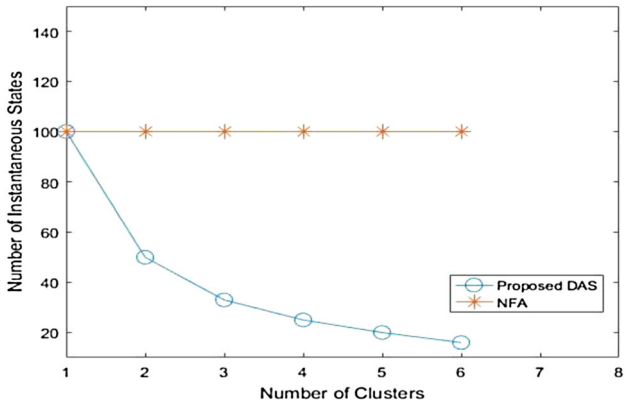

Fig. 2 Tradeoff between number of clusters and number of instantaneous states

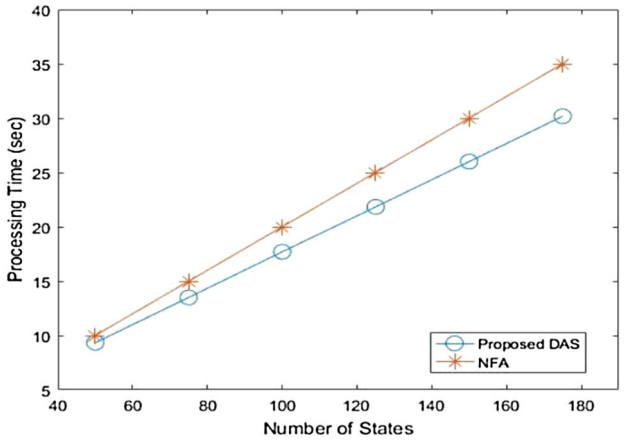

Fig. 3 Tradeoff between number of instantaneous states and processing time

automaton should be big O (m). In DAS model the numbers of all possible states are partitioned by H different clusters and h denotes cluster selection time. So each cluster handled only subset of m possible states and its transactions are manageable easily. Therefore DAS State Space Complexity is big O (|Q||m|). Where N and S are denote as maximum number of states in NFA and DAS respectively.

Figure 2 shows that, the number of instantaneous states representing DAS are decreasing when number of clusters increased at N = 100. In Non-Deterministic Finite Automata (NFA) system representation the number of instantaneous states are not reduces, it may cause more time and space complexity in large real time environments

Figure 3 shows the relationship between instantaneous states of DAS system with processing time in secs when h = 1 s, N = 200, and H = 6. It clearly indicates proposed DAS system requires lesser processing time compared to NFA system.

6 Conclusion

In this paper, we proposed DAS model to build intelligent control system for AWSNs with the use of distributed automaton. Proposed model is capable to detect various attacks. Such as, DoS, replay, control hijacking, selfish node, sinkhole, wormhole, and hello flooding attacks with low computational overheads. Because of considering time and location constraints are an attribute to define distributed automaton and DAS. Every successful transition in DAS is performed by verifying time and location information, it makes every intrusion activity difficult to proliferate in DAS architecture. Through asymptotic analysis, we proved that our DAS model utilizes less state space to accommodate entire behaviour of the network.

References

1. Xie, M., Han, S., Tian, B., Parvin, S.: Anomaly detection in wireless sensor networks: a survey. J. Netw. Comput. Appl. **34**(4), 1302–1325 (2011)
2. Mahmood, M.A., Seah, W.K.G., Welch, I.: Reliability in wireless sensor network: a survey and challenges ahead. Comp Netw. 1–55 (2014)
3. Newsome, J., Shi, E., Song, D., Perrig, A.: The Sybil attack in sensor networks: analysis & defenses. In: Proceedings of the 3rd International Symposium On Information Processing In Sensor Networks. April 26–27. (2004)
4. Raymond, D.R., Midkiff, S.F.: Denial of service in wireless sensor network: attacks and defenses. IEEE Pervasive Comput. **7**(1), 74–81 (2008)
5. Yi, P., Dai, Z., Zhong, Y., Zhang, S.: Resisting flooding attacks in adhoc networks. In: Proceeding Int'l Conf. Information Technology: Coding and Computing, vol. 2, pp. 657–662. (2005)
6. Parno, B., Perrig, A., Gligor, V.: Distributed detection of node replication attacks in sensor networks. In: Proceedings of IEEE Symposium on Security and Privacy, pp.49–63 (2005)
7. Hu, Y.C., Perrig, A., Johnson, D.B.: Packet leashes: a defense against wormhole attacks in wireless adhoc networks. In: Proceedings of IEEE Infocomm (2003)
8. Michiardi, P., Molva, R.: Core: a collaborative reputation mechanism to enforce node cooperation in mobile adhoc networks. In: Advanced Communications and Multimedia Security.

IFIP, The International Federation for Information Processing, vol. 100. Springer (2002)
9. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: attacks and countermeasures. Adhoc Netw. **1**, 293–315 (2003)
10. Alrajeh, N.A., Khan, S., Shams, B.: Intrusion detection systems in wireless sensor networks: a review. Int. J. Distrib. Sens. Netw. **2013**, 1–7 (2013)
11. Vasilomanolakis, E., Fischer, M., Max, M., Ebinger, P., Kikiras, P., Schmerl, S.: Collaborative intrusion detection in smart energy grids. In: Proceedings of the International Symposium for ICS & SCADA Cyber Security, Electronic Workshops in Computing (eWiC). pp. 97–100. (2013)
12. Lei, Y., Zhang, Y., Zhao, Y.: The research of coverage problems in wireless sensor network. In: Proceedings of International Conference On Wireless Networks And Information Systems, pp. 31–34. (2009)
13. Yavuz, F., Zhao, J., Yagan, O., Gligor, V.:On secure and reliable communications in wireless sensor networks: towards k-connectivity under a random pairwise key pre distribution scheme. In: IEEE International Symposium On Information Theory. pp. 2371–2375 (2014)
14. Kaplantzis, S., Shilton, A., Mani, N., Ahmet, Y.:Detecting selective forwarding attacks in wireless sensor networks using support vector machines: In: 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) pp. 335–340 (2007)
15. Roman, R., Zhou, J., Lopez, J.:Applying intrusion detection systems to wireless sensor networks. In: Proceeding of the 3rd IEEE Consumer Communications and Networking Conference (CCNC), vol. 1, pp. 640–644. (2006)
16. Huang, Y., Fan,W., Lee, W., Yu, P.S. (2003) Cross-feature analysis for detecting adhoc routing anomalies. In: Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 478–487. (2003)
17. Conti, M., Giordano, S.: Mobile adhoc networking: milestones, challenges, and new research directions. IEEE Commun. Mag. **52**(1), 85–96 (2014)
18. Harrop, P.:Wireless sensor networks and the new Internet of things. Energy Harvest J. http://www.energyharvestingjournal.com (2012)
19. Rawat, P., Deep, K., Chaouchi, H., Marie, J.: Wireless sensor networks: a survey on recent developments and potential synergies. J Supercomput. **68**, 1–48 (2014)
20. Gungor, V.C., Lu, B., Member, S., Hancke, G.P., Member, S.: Opportunities and challenges of wireless sensor networks in smart grid. IEEE Trans. Ind. Electron. **57**(10), 3557–3564 (2010)
21. Ferdoush, S., Li, X.:Wireless sensor network system design using raspberry pi and arduino for environmental monitoring Applications. In: The 9th International Conference on Future Networks and Communications, Procedia Computer Science, vol. 34, pp. 103–110. (2014)
22. Gutie, J.A.: On the use of IEEE std. 802. 15. 4 to enable wireless sensor networks in building automation. Int. J. Wirel. Inf. Netw. **14**(4), 295–301 (2007)
23. Abbasi, A.Z., Islam, N., Shaikh, Z.A.: A review of wireless sensors and networks' applications in agriculture. Comput. Stand. Interfaces **36**(2), 263–270 (2014)

S. Venkatraman received the B.E. degree in Computer Science and Engineering from Bharathidasan University, Tiruchirappalli, India in 2001 and the M.E. degree in software engineering from Anna University, Chennai, India in 2004. His research interests include Cybernetics, Network Security, Internet of Things, Machine Learning, Currently, he is pursuing his Ph.D. degree in the CSE Dept. at NIT Puducherry, India.

P. Arun Raj Kumar received his B.E. degree in Computer Science and Engineering from Malaviya National Institute of Technology, Jaipur, India in 2002 and the M.E., and Ph.D. degrees in Computer Science and Engineering from National Institute of Technology, Tiruchirappalli, India in 2008, and 2013, respectively. His research interests include Network Security, Computer Networks, Machine Learning, and Internet of Things. He is currently Assistant Professor in the Dept. of CSE at National Institute of Technology Puducherry, India.