iTrust: An Integrated Trust Framework for Wireless Sensor Networks

Kuldeep Yadav
Indraprastha Institute of Information Technology
New Delhi, India-110078
kuldeep@iiitd.ac.in

Avinash Srinivasan
Bloomsburg University
Bloomsburg, PA 17815
avinash@bloomu.edu

ABSTRACT

Designing security solutions for Wireless Sensor Networks is a challenging task due to the potential hostile and unattended environment in which they operate as well as their resource constrained nature. A trust management framework can be useful for detecting untrustworthy nodes under such operational conditions. In an unattended autonomous network, the attacker can capture a sensor node and modify its regular functioning. Consequently, the compromised node will thereafter behave erratically, which, in most cases, is observable by nodes in the corresponding neighborhood. In this paper, we propose *iTrust*- an integrated trust framework in which monitor nodes, a set of specialty nodes, will evaluate neighborhood nodes based on their behavior in a session wise manner. Monitor nodes, in promiscuous mode, will garner information about nodes in their neighborhood. After each session, they will share trust indices of each node with their neighbors, which is used for future decision-making. We have simulated *iTrust* framework with a tolerance of 5%-25% network error rate and evaluated its performance. We have further evaluated the attack detection effectiveness of *iTrust* framework by simulating different attack scenarios and confirmed its robustness to several known attacks.

Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: General- Security and protection.
C.2.1 [Network Architecture and Design]: Wireless communication.
C.2.3 [Network Operations]: Network Monitoring.
I.2.9 [Robotics]: Sensors.

General Terms

Security, Design, Performance and Reliability.

Keywords

Monitoring, Promiscuous, Security, Trust, and Sensor Networks.

SAC'10, March 22-26, 2010, Sierre, Switzerland.
Copyright 2010 ACM 978-1-60558-638-0/10/03…$10.00.

1. INTRODUCTION

Wireless Sensor Networks (WSNs), composed of sensor nodes, are being used in diversified application domains like *military surveillance*, *agricultural farming*, *traffic management*, *habitat monitoring*, *forest fire detection*, etc. Security in WSNs, which carry sensitive data and operate autonomously in unattended hostile environments, is therefore very critical. However, the security requirements in WSNs is considerably different from other networks such as MANETS due to the following reasons-

1. Sensor nodes are highly resource constrained with limited battery life, low bandwidth, small processing capabilities, and memory constraints. Consequently, computationally intensive security protocols cannot be employed to secure a WSN.
2. Sensor nodes are vulnerable to node compromise attack, which subsequently leads to malicious misbehavior of compromised nodes.
3. Sensor nodes in a network are also vulnerable to selfish misbehavior arising due to scarcity of resources.
4. Additionally, network errors are very high in WSNs, particularly due to the environment in which they operate.
5. Cryptography and authentication alone are not sufficient to address all the security requirements due to aforementioned unique characteristics as well as planned attacks by an adversary.
6. Finally, each application domain requires a different level of security. Hence a general security framework may have to be fine tuned to address the particular security requirement of each application domain.

In this ongoing work, we are designing *iTrust*- an integrated trust framework that can detect most known attacks. In *iTrust*, one set of nodes called monitor nodes, with a watchdog mechanism, monitor all nodes in their neighborhood based on various tasks and service, and maintain a reputation table with trust indices of all neighborhood nodes. Note that, in *iTrust*, even sensor nodes monitor the neighborhood, but only in visible mode and not in promiscuous mode. We will be discussing this in further details in later sections. Various trust and reputation based schemes [3, 4, 5, and 6] have been recently proposed for WSN. Generally, all solutions proposed thus far suffer from a common limitation- they have been designed for a specific protocol layer or service such as routing, data aggregation, etc., and fail to differentiate between various services. Also, sensor networks are highly prone to network errors, and therefore behavior based trust frameworks are at times incapable of distinguish a planned attack from a erratic behavior arising from high network error rate. Our framework, *iTrust*, provides a distributed, scalable, and a generalized approach to detect attacks and isolate misbehaving nodes. We have also shown its effectiveness and efficiency in attack detection through simulation studies. The proposed *iTrust* framework is a holistic solution in that it addresses the security concerns from a system perspective rather than a layered approach.

The contributions of this paper can be summarized as follows-

1. In *iTrust*, not all nodes are required to operate in promiscuous mode to monitor the neighborhood behavior. Only the special monitor nodes operate in promiscuous mode. This helps in conserving precious sensor node resources.
2. We are employing only a subset of nodes, hereinafter referred to as monitor nodes, for monitoring neighborhood behavior. This helps in extending the network lifetime.
3. Our *iTrust* framework is adaptable to environmental conditions with a learning phase (first session) that dynamically determines and sets the threshold for trust values.
4. Finally, we have tried to include most parameters, layered as well as cross layer, for detection of attacks thus not restricting it to specific attacks as in [7, 10, 11].

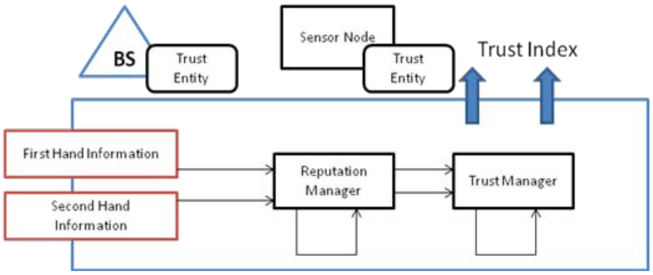

Figure 1: Generalized Block diagram for trust framework in Wireless Sensor Networks.

2. RELATED WORKS

In this section, we will review articles that are most closely related to our work due to paper length restrictions. Previously, most works in this area have focussed in building intrusion detection, prevention, and tolerance [7,8,9]. But now it is slowly moving towards building trust models since a robust trust model inherently works as a intrusion detction scheme as well. A complete and thorough discussion of all intrusion detection schemes is beyond the scope of this paper.

We now briefly discuss the limitations of some select and closely reated existing trust models. There are mainly two kind of approaches followed in designing trust models-

1. *Centralized*- In this approach, a resourceful entity like a Base Station is used for monitoring and to maintain trust indices of all the nodes in a network.
2. *Distributed* [4] – In this approach, either a node maintains its neighbors' trust indices or a super node such as a cluster head or a monitor node is delegated this responsibility.

Some of the proposed trust models work for a specific service or attack like secure localization, secure routing [4], secure data aggregation etc. Schemes proposed by [4] forces all nodes to maintain trust indices for their neighboring nodes. Maintaing trust at each node will decrease a node's life time and eventually shortens the entire network lifetime. This may not be suitable for some WSN application scenarios as they require a prolonged network lifetime. Also, in existing trust models there is lack of attack detection effectiveness as they are using very few paremeters for calculating trust indices. To address the sensor malfunction problems, Ganeriwal and Srivastava [3] have proposed a framework in which sensor nodes maintain reputation for other nodes in the network. A sensor node continuously builds these reputation metrics for other nodes by monitoring their behavior and rating them as being cooperative (expected behavior of the nodes in the network) or non-cooperative (unexpected behavior that is most likely the result of a system fault or node compromise). Finally, the node uses this reputation to evaluate the trustworthiness of other nodes and the data they provide.

In [12] some rules are proposed on the basis of which intrusions can be detected. There are 3 phases of this protocol- Phase-1 is data acquisition, Phase-2 is rule application, and Phase-3 is intrusion detection. Phase-1 is the analysis phase when the number of failures is compared to the expected amount of occasional failures in the network. It also takes network errors into account. But the main problem with this solution is that it uses a small number of trust parameters, which can restrict its application to some certain types of attacks. Consequently, network lifetime will suffer in [12]. To address the major issues of a trust model, we have focussed on designing *iTrust*- an integrated trust framework to detect most known attacks attacks and isolate misbehaving nodes.

3. PROBLEM FORMULATION

Trust represents the cumulative performance of the node in past tasks and decides the role of the node in future tasks [5]. Trust Management can be defined as a process of monitoring activities in a network system, which can be done by either the sensor nodes or a small set of special nodes, such as monitor nodes. The monitor nodes collect activity information and then analyze these data to determine whether or not there are any activities that violate the security rules. We have considered designing an integrated trust framework in the context of following design goals.

1. Decentralized Implementation

2. Fast attack detection and accuracy under high network error scenarios.

3. Combination of direct (first hand) and indirect (second hand) trust.

4. Trust associated with past behavior should be used cautiously in future network decisions.

5. Framework should work for all protocol layers and services- a holistic solution.

Attacks that can be confused with network errors such as *packet collision* and *packet loss* include physical attacks like *radio jamming*. Whereas, *selective forwarding*, *hello attacks*, *flood attacks* are not mistaken for any inherent network errors. We have made the following assumptions while implementing this protocol.

1. Network consists of two types of nodes- regular sensors and monitor nodes.

2. Sensor nodes are assumed to be static and can operate only in visible mode.
3. Monitor nodes can operate in promiscuous mode as well as visible mode.
4. A basic crypto infrastructure already exists before implementing the *iTrust* framework.
5. There is no intruder present during network deployment and for a very short interval, say δ, thereafter. δ is a tunable parameter.

4. SOLUTION MODEL FOR DEFINED PROBLEM

Due to space limitations, we are hereby discussing only the important steps of *iTrust* framework. For trust management, we explore a distributed model using monitor nodes that are assigned an extra responsibility to store trust parameter values. Each node in *iTrust* can operate in one of to modes- *visible mode* and *promiscuous mode*. The energy consumption in visible mode is many folds lower compared to promiscuous mode. Therefore, only monitor nodes will operate in promiscuous mode, when necessary, to ensure that we conserve precious sensor node energy for intended network services as far as possible. This also helps in prolonging the network lifetime. Both sensor and monitor nodes will *wake-up* and *sleep-on-idle* according to the underlying scheduling algorithm, a detailed discussion of which is beyond the scope of this paper. However, there is an exception to monitor nodes in this aspect. They switch to promiscuous mode on idle instead of sleeping. This is to facilitate monitoring of the neighborhood. Additionally, monitor nodes operating in promiscuous mode maintain trust indices for all nodes in their neighborhood. However, other nodes are unaware of the promiscuous listening capabilities of the monitor nodes and expect to be evaluated on parameters that do not require promiscuous listening. This ensures that nodes are behaving normally without altering their behavior conscious of being evaluated. We have considered parameters from all layers to isolate nodes having a lower trust index. Following is the complete list of parameters which is used for evaluating trust.

1. Available Energy (AE).
2. Packet Signal Strength (PSS).
3. Control Packets Received for Forwarding (CPRF).
4. Control Packets Forwarded (CPF).
5. Number of Packets Transmitted (NPT).
6. Number of Packet Collisions (NPC).
7. Data Packets Received for Forwarding (DPRF).
8. Data Packets Received and Forwarded (DPF).
9. Number of Packets Dropped (NPD).
10. Number of Packets Received (NPR).
11. Number of Route Requests Sent (NRRS).
12. Number of Route Replies Received (NRRR).
13. Number of Packets for which MAC Failed (NPMF).
14. Number of Packets for which Decryption Failed (i.e. non meaningful text) (NPDF).
15. Number of Packets for which Freshness Check Failed (NPCF).
16. Number of Beacon Packets Sent (NBPS).
17. Number of Beacon Packets Dropped (NBPD).
18. Number of Hello Packets Sent in a Session (NHPS).
19. Number of Data Reading Packets Received (NDPR).

Here parameters' values are collectively used for detecting and isolating attacker nodes. For instance, parameters *Available Energy* (AE) and *Packet Signal Strength* (PSS) can be used for detecting laptop class attackers. Because AE is a decreasing entity, which should decrease as time increases, so should PSS. The following are the main steps necessary in core functioning of our *iTrust* framework.

Step1: Neighbor discovery phase is the first step in *iTrust*. During this phase, nodes get information about their neighbors' ID and their current energy values.

Step 2: This step can also be referred to as learning phase. This step starts in conjunction with neighbor discovery phase. During this phase, monitor nodes collect above listed parameters by promiscuous listening. Monitor nodes will calculate a trust index with these parameter values for each sensor node in their neighborhood. We have already specified our assumption that there is no intruder present in a network during network deployment and for a short duration δ thereafter. So during this phase, monitor nodes will learn as to what could be the possible trust value if a node behaves honestly/dishonestly in a network. This phase is important because sometimes attacks are confused with network errors. After the learning phase, all monitor nodes share their trust regulation table containing trust indices of nodes in their respective neighborhoods with each other and publishes it in the network.

Step 3: After the learning phase, each monitor will get to know the normal trust values of each sensor node in its neighborhood. Monitor nodes will post their values to nearby neighborhood at the end of each session for reference in further communication. If a node's trust index deviates from normal trust index, it can be reported to the base station. A sensor node can also report if a monitor node's trust index, which it computes in the visible mode, is below the acceptable threshold. We have used weighted trust, which is calculated as follows for a sensor node *i*-

$$T_j(i) = M_j(i) \times W_m + S_j(i) \times W_s$$

Here $M_j(i)$ represents the trust index of node *j* that the monitor node is reporting to node *i.* If there is more than one monitor node reporting on node *j*, then take the average of the reported trust indices. Similarly, $S_j(i)$ represents the trust index of node *j* maintained by node *i* itself. W_m and W_s are the weights associated with trust indices reported by monitor nodes and computed by sensor nodes respectively. Generally sensor nodes give more weight to trust indices sent by monitor nodes, i.e., usually, $W_m > W_s$. The sum of different weights will be 1 and certainly both factor values will also have a value of less than 1. Finally, weighted trust for a node will range between 0 and 1. For the above parameters, trust will be calculated layer wise and after computation a combined trust based on trust values of all layers is calculated. Application layer trust will be computed as following-

$$A_1 = (AE_i(T1) - AE_i(T2)) / AE_i(T1) \qquad \text{where } T1 < T2.$$

$A_2 = (PSS_i\,(T1) - PSS_i\,(T2)) / PSS_i\,(T1)$ where $T1 < T2$.

$A_{3\,=}\ 1 - NHPR_i / NPR_{i.}$

$A_{4\,=}\ 1 - NPFC_i / NDR_{i.}$

So, the trust will be calculated as Tapp = w1* $A_{1\,+}$ w2* A_2 + w3* A_3 + w4* A_4. Here w1+w2+w3+w4=1. After calculting trust of other three layers, combined trust is computed using the following equation-

T_j (i) = Tapp * WEIGHT_APP + Troute * WEIGHT_ROUTING + Tmac * WEIGHT_MAC + Twireless + WEIGHT_WIRELESS.

After calculating trust with the above equation, we have to combine direct trust with indirect trust and presession trust. We have to consider presession trust to have consistency in the trust frmework,

$\mathbf{T^f_j\,(i)}$= Tpi * WEIGHT_PRE_SESSION + T_j (i) * WEIGHT_CURRENT_SESSION

Here, Tpi is the presession trust and Ti is current session trust. Combination of both these will give the final trust T^f_j for a node *j* computed by node *i*. A node's trust index should reflect the behavior of the node for ist entire life upto the current point. However, current actions should be more dominant in the computed trust index, giving them an opportunity to rectify their past behavior. Therefore, weight assigned to current session trust will always be higher because of its freshness.

5. SIMULATION RESULTS

There is a lack on simulator for Wireless Sensor Networks Security. Current simulation tools do not allow modeling of attacker nodes. Therefore, we have extended OMNeT++ [15] based Castalia simulator [16] to adapt and code this functionality. The implementation of this framework is done in such a way that it could work for any number of sensor nodes and any number of malicious nodes can be introduced into the network. Table-1 has some important parameter for simulation setup in Castalia. For further information, refer to the corresponding .ini file in Castalia.

Table 1: Simulation Parameters

Radio	TelosB_CC2420
Session Time	50 simulation seconds
Initial Energy	29160 J (2AA Batteries)
Wireless Channel	Path loss exponent=2.4 with all bidirectional links and additive interference model [17].

For testing and analysis purpose, we have deployed 25 nodes with a 5x5 grid as the underlying network topology. Since our protocol is a distributed implementation, we believe scalability will not been an issue. In the 25 nodes, 4 nodes are assigned the responsibility of monitor nodes so that the entire network can be covered. As described earlier, the first phase is the learning phase for the protocol during which period it will set up threshold values for low and high trust indices.

After the learning phase, the nodes will setup the following table, which can be referenced for detecting an attacker. Based on Table-2, we have tested attack detection effectiveness of this protocol. Our framework has been evaluated for a network error rate of 5%-25%. We have tested its attack detection effectiveness by simulating some popular attacks like collision attack, selective forwarding attack, etc.

Table 2: Trust Regulation Table

Trust Level	Description	Range
1.	Low Trust	{0,0.59}
2.	Medium Trust	{0.6,0.79}
3	High Trust	{0.8,1}

The trust values of all nodes in the network with a network error rate of 5% are plotted in Fig. 2 for two sessions. We can see from the results that the average trust value of node with a network error rate of 5% for session-1 is approximately 0.975 with a high of 0.99 (node IDs- 6, 8, 16, 18) and a low of 0.96 (node IDs- 9, 11). The trust values of all nodes range between 0.96 - 0.99 for session-1. Similarly, for session-2, the average trust value of node with a network error rate of 5% is approximately 0.965 with a high of 0.99 (node IDs- 6, 8, 16, 18) and a low of 0.942 (node ID- 7). It is evident from the results that the average trust value of nodes slightly decreases with time, i.e., the value decreases slightly from session-1 to session-2. This is acceptable since node behavior and subsequently the network stabilize over time. The trust values of all nodes range between 0.942 - 0.99 for session-2.

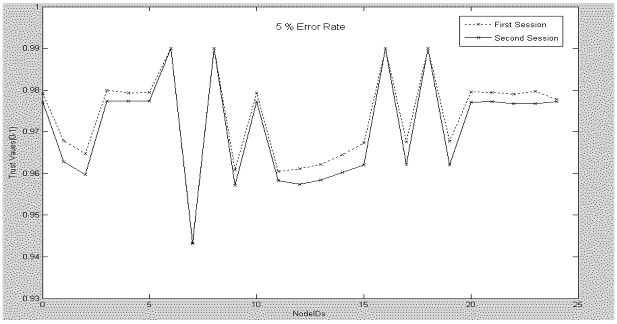

Fig. 2: Trust index values of all nodes with 5% network error rate.

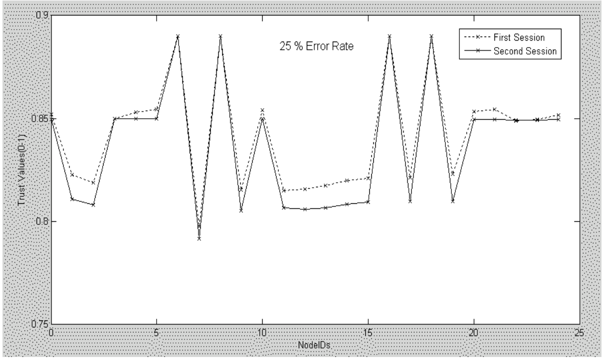

Fig. 3: Trust index values of all nodes with 25% network error rate.

Similarly in Fig. 3 we have plotted the results showing the trust values of all nodes in the network with a network error rate of 25%. We can see from the results that the average trust value of node with a network error rate of 25% for session-1 is approximately 0.84 with a high of 0.89 (node IDs- 6, 8, 16, 18) and a low of 0.8 (node ID- 7). Similarly, for session-2, the average trust value of node with a network error rate of 25% is approximately 0.83 with a high of 0.89 (node IDs- 6, 8, 16, 18) and a low of 0.79 (node ID- 7). Here again, the average trust value of nodes slightly decreases with time, i.e., the value decreases slightly from session-1 to session-2.

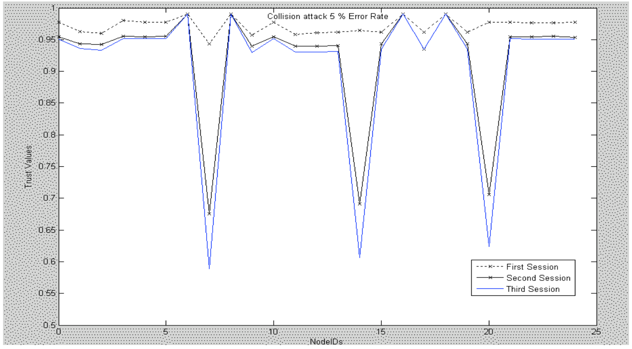

Fig. 4: Attack detection effectiveness for Collision Attack with 5% network error rate.

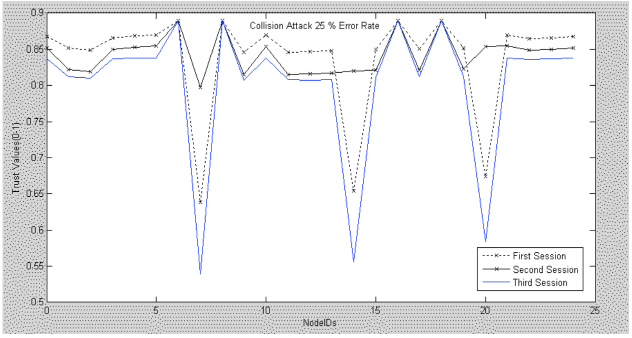

Fig. 5: Attack detection effectiveness for Collision Attack with 25% network error rate.

In Fig. 4, we have presented the detection efficiency of the network against *collision attacks* with 5% network error rate. We have measured the drop in trust indices of each node in the network when simulating the network for this scenario. We have assumed that three nodes (Node ID- 7, 14, 20) started to violate transmission policy of the network thereby launching collision attack. It is clear from Figs. 4 and 5 that as network error increases, our framework was able to detect misbehaving nodes after two sessions. Therefore, we can infer that trust indices of misbehaving nodes will decrease as the number of session increases. So we can isolate misbehaving nodes after some time. Trust indices for nodes could be used for future network decisions like routing, cluster head election, etc.

In Fig. 6, we have presented the detection efficiency of the network against *selective forwarding attacks* with 25% network error rate. We have measured the drop in trust indices of each node in the network when simulating the network for this scenario. Finally, in Fig. 7 we have presented the detection efficiency of the network against *MAC layer attacks* with 25% network error rate. We have measured the drop in trust indices of each node in the network when simulating the network for this scenario. Due to paper length constraints we could not discuss further details on results in this paper.

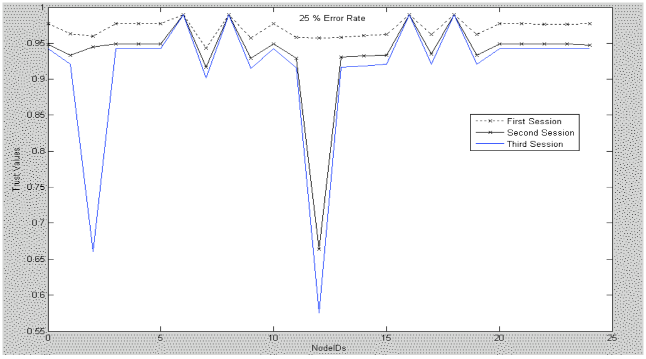

Fig. 6: Attack detection effectiveness for Selective forwarding attack with 25% Network error.

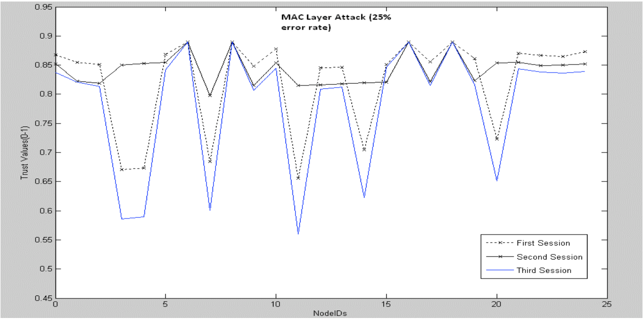

Fig. 7: Attack detection effectiveness for Mac Layer (Hello flood) attack with 25% Network error.

6. CONCLUSION AND FUTURE WORK

In this work, we have mainly focused on designing and implementing *iTrust*- an adaptive trust framework for WSNs. As WSNs have highly diversified application domains, static trust mechanisms will neither be effective nor efficient. So, by using learning phase after network deployment, we can build a trust regulation table which can be used as the baseline threshold for detecting attacks. We have also taken network errors into account, which are sometimes confused with attacks. Through simulation studies, we have checked the attack detection effectiveness of *iTrust* for three different attacks. As part of our future work, we will evaluate the attack detection effectiveness of *iTrust* by simulating more attack scenarios. We are also planning to extend *iTrust* to the domain of mobile sensor networks. We would also like to study the impact of compromising monitor nodes on the performance of *iTrust*.

7. ACKNOWLEDGMENTS

This work has been supported in part by the Bloomsburg University Research and Disciplinary Grants 2009-2010. Our special thanks to Castalia [16] developers for developing such a customizable and adaptable simulator. We are also thankful to Vinayak Naik for his valuable suggestions in the implementation of this framework.

8. REFERENCES

[1] Kuldeep, Kalpana Sharma, M.K. Ghose," Wireless Sensor Networks Security: A New Approach", *In proceedings of 16th International Conference on Advanced Computing and Communication*,13-16 Dec 2008, MIT Chennai.

[2] Kuldeep et. al, "Complete Security Framework for Wireless Sensor Networks", *In International Journal of Computer Science and Information Security (IJCSIS)* , Vol 3, No. 1, 2009, ISSN 1947-5500.

[3] S. Ganeriwal and M. B. Srivastava. Reputation-Based Framework for High Integrity Sensor Networks. In *2nd ACM Workshop on Security of Ad Hoc and Sensor Networks.* pages 66–77, Washington, DC, USA,2004.

[4]. Z.Yao, D. Kim, I. Lee, K. Kim, and J. Jang. A Security Framework with Trust Management for Sensor Networks. *In Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks*, pages 190–198, 2005.

[5] Lei Huang, Lei Li and Qiang Tan, Behavior-Based Trust in Wireless Sensor Network**,** H.T. Shen et al. (Eds.): APWeb Workshops 2006, LNCS 3842, pp. 214 – 223, 2006.

[6] Haiguang Chen, Task-based Trust Management for Wireless Sensor Networks**,** *International Journal of Security and Its Applications*,Vol. 3, No. 2, April, 2009.

[7] Q. Ren and Q. Liang. Secure media access control (mac) in wireless sensor networks: intrusion detections and countermeasures. In PIMRC 2004, 2004.

[8] Doumit, S. and Agrawal, D. (2003) Self-organized critically and stochastic learning based intrusion detection system for wireless sensor networks. MILCOM.

[9] Onat, I,Miri, A, "An Intrusion Detection System for Wireless Sensor Networks" In proceedings of WiMob' 2005, 22-24 Aug 2005, pp(253-259).

[10] Bo Yu and Bin Xiao. Detecting Selective Forwarding Attacks in Wireless Sensor Networks. In Parallel and Distributed Processing Symposium, 2006. IPDPS (2006).

[11]. Y.-C. Hu, A. Perrig, and D. B. Johnson, Packet leashes: A defense against wormhole attacks in wireless networks, in Proc of IEEE Infocomm (2003).

[12] Haiguang Chen et. al **, "**Lightweight Anomaly Intrusion Detection in Wireless Sensor Networks", C.C. Yang et al. (Eds.): PAISI 2007, LNCS 4430, pp. 105–116, 2007.

[13] Lei Huang, Lei Li, and Qiang Tan, Behavior-Based Trust in Wireless Sensor Network, H.T. Shen et al. (Eds.): APWeb Workshops 2006, LNCS 3842, pp. 214 – 223, 2006.'

[14] A. Srinivasan and J. Wu. "A Survey on Secure Localization in Wireless Sensor Networks". In Encyclopedia of Wireless and Mobile Communications, B. Furht (ed.), CRC Press, Taylor and Francis Group. 2007.

[15] www.omnetpp.org

[16] http://castalia.npc.nicta.com.au/

[17] Athanassios Boulis, "Castalia: User' Manual" August 2009, NICTA.

Biography

Kuldeep Yadav- Currently, he is a PhD Scholar at IIIT Delhi. Previously, he has completed B.Tech from Sikkim Manipal Institute of Technology and was ranked second in University. His current research interests include Mobile P2P and security in wireless sensor networks. He has published over 5 research papers in International Conference and Journals. He has also served as external reviewer for International Conferences and has organized several programming and paper presentation contests. He has been the recipient of HiPC student travel fellowship for the last two years.

Avinash Srinivasan- Dr. Avinash Srinivasan received his B.E. (1999) in Industrial Engineering from University of Mysore, India, MS (2003) in Computer Science from Pace University, New York, USA, and Ph.D. (2008) in Computer Science from Florida Atlantic University, Florida, USA. He joined Bloomsburg University as an Assistant Professor of Computer Forensics in Aug. 2008. His research interests are wireless and sensor network security, reputation and trust-based security models for wireless and sensor networks, and digital forensics. Dr. Srinivasan has published over 18-refereed articles in international journals and conferences. He is currently serving as an Associate Editor for Wiley and Son's Security and Communication Networks journal. He has served on the Program Committee of over 45 international conferences and workshops and reviewed for numerous international journals including Transactions on Wireless Communications, Mobile Networks and Applications, and Journal of Parallel and Distributed Computing. Dr. Srinivasan has been featured in Who's Who in America 2010.