Cryptography via Virtual Energy for Wireless Sensor Networks

K. Naga Krishnaja

Abstract—**According to the survey on wireless sensor networks several sensors that perform only sensing can be deployed. The positions of the sensors and communications topology are carefully engineered. They transmit time series of the sensed phenomenon to the central nodes where computations are performed and data are fused. Sensor network protocols and algorithms must possess self-organizing capabilities. Another unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are fitted with an on-board processor. Instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.**

Index Terms—**Cost-efficient, encryption, Keying, Virtual energy.**

I. Introduction

WSN technology will be used in a variety of application scenarios. Reference [1] mentions the typical application areas of wsn's include environmental, military, and commercial enterprises. For example, in a battlefield scenario, sensors may be used to detect the location of enemy sniper fire or to detect harmful chemical agents before they reach troops. In another potential scenario, sensor nodes forming a network under water could be used for oceanographic data collection, pollution monitoring, assisted navigation, military surveillance, and mine reconnaissance operations. Future improvements in technology will bring more sensor applications into our daily lives and the use of sensors will also evolve from merely capturing data to a system that can be used for real-time compound event alerting.

Protocols should be resilient against false data injected into the network by malicious nodes. Otherwise, consequences for propagating false data or redundant data are costly, depleting limited network resources and wasting response efforts. However, securing sensor networks poses unique challenges to protocol builders because these tiny wireless devices are deployed in large numbers, usually in unattended environments, and are severely limited in their capabilities and resources (e.g., power, computational capacity, and memory)[2]. For instance, a typical sensor operates at the frequency of 2.4 GHz, has a data rate of 250 Kbps, 128 KB of program flash memory, 512 KB of memory for measurements, transmit power between 100 micro W and 1 m W, and a communications range of 30 to 100 m. Therefore, protocol builders must be cautious about utilizing the limited resources onboard the sensors efficiently.

II. Key Scheme

A. Sensor Networks Communication Architecture

The sensor nodes are usually scattered in a sensor field as shown in Figure 1. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink and the end users. Data are routed back to the end user by a multi-hop infrastructure-less architecture through the sink as shown in Figure1.

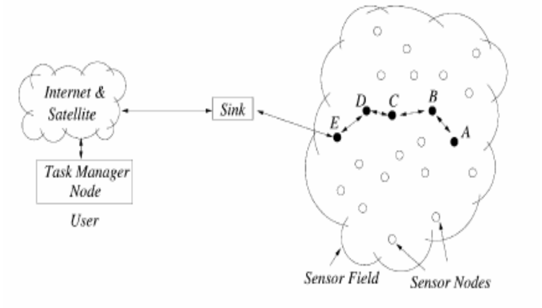


Fig. 1. Sensor nodes scattered in sensor fields

The sink may communicate with the task manager node via Internet or Satellite. The protocol stack used by the sink and all sensor nodes is given in Figure 2.

This protocol stack combines power and outing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium, and promotes cooperative efforts of sensor nodes.

The protocol stack consists of the application layer, transport layer, network layer, data link layer physical layer, power management plane, mobility management plane, and task management plane. Depending on the sensing tasks, different types of application software can be built and used on the application layer. The transport layer helps to maintain the flow of data if the sensor networks application requires it. The network layer takes care of routing the data supplied by the transport layer. Since the environment is noisy and sensor nodes can be mobile, the MAC protocol must be power aware and able to minimize collision with neighbors broadcast. The physical layer addresses the needs of a simple but robust modulation, transmission and receiving techniques.

In addition, the power, mobility, and task management planes monitor the power, movement, and task distribution among the sensor nodes. These planes help the sensor nodes coordinate the sensing task and lower the overall power consumption. The power management plane manages how a sensor node uses its power. The mobility management plane detects and registers the movement of sensor nodes, so a

Manuscript received February 5, 2012; revised February 25, 2012. This work was supported in part by the IT Department of JNTUK, Vzm.

K. Naga Krishnaja is with the JNTUK VZM in Information and Technology, Andhra Pradesh, India (e-mail: nagakrishnaja@gmail.com).

route back to the user is always maintained, and the sensor nodes can keep track of who are their neighbor sensor nodes. The task management plane balances and schedules the sensing tasks given to a specific region.

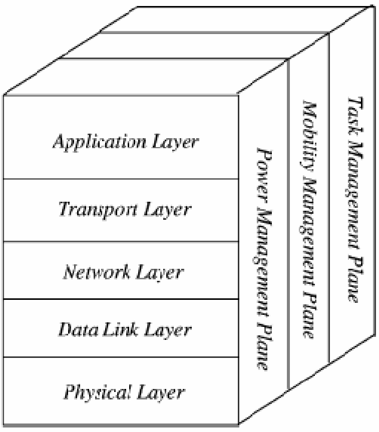


Fig. 2. The sensor networks protocol stack

B. Related Work

En-route dynamic filtering of malicious packets has been the focus of several studies, including dynamic en-route filtering (DEF) by Yu and Guan [3], statistical en-route filtering (SEF) [2], and Secure Ticket-Based En-route Filtering (STEF) [4] where they were compared with the VEBEK framework, the reader is referred to that section for further details as not to replicate the same information here. Moreover, Ma's work [5] applies the same filtering concept at the sink and utilizes packets with multiple MACs appended. A work [6] proposed by Hyun and Kim uses relative location information to make the compromised data meaningless and to protect the data without cryptographic methods. In [7], using static pair-wise keys and two MACs appended to the sensor reports, "an interleaved hop-by-hop authentication scheme for filtering of injected false data" was proposed by Zhu et al to address both the insider and outsider threats. However, the common downside of all these schemes is that they are complicated for resource-constrained sensors and they either utilize many keys or they transmit many messages in the network, which increases the energy consumption of WSNs. Also, these studies have not been designed to handle dire communication scenarios unlike VEBEK. Another significant observation with all of these works is that a realistic energy analysis of the protocols was not presented. Lastly, the concept of dynamic energy-based encoding and filtering was originally introduced by the DEEF [8] framework. Essentially, VEBEK has been largely inspired by DEEF. However, VEBEK improves DEEF in several ways. First, VEBEK utilizes virtual energy in place of actual battery levels to create dynamic keys. VEBEK's approach is more reasonable because in real life, battery levels may fluctuate and the differences in battery levels across nodes may spur synchronization problems, which can cause packet drops. Second, VEBEK integrates handling of communication errors into its logic, which is missing in DEEF. Lastly, VEBEK is implemented based on a realistic WSN routing protocol, i.e., Directed Diffusion [9], while DEEF articulates the topic only theoretically. Another crucial idea of this chapter is the notion of sharing a dynamic cryptic credential (i.e., virtual energy) among the sensors. A similar approach was suggested inside the SPINS study [10] via the SNEP protocol. In particular, nodes share a secret counter when generating keys and it is updated for every new key. However, the SNEP protocol does not consider dropped packets in the network due to communication errors. Although another study, Minisec [11], recognizes this issue, the solution suggested by the study still increases the packet size by including some parts of a counter value into the packet structure. Finally, one useful pertinent work [12] surveys cryptographic primitives and implementations for sensor nodes.

C. Modules

This framework is comprised of three modules: Virtual Energy-Based Keying, Crypto, and Forwarding. The virtual energy-based keying process involves the creation of dynamic keys. Contrary to other dynamic keying schemes, it does not exchange extra messages to establish keys. A sensor node computes keys based on its residual virtual energy of the sensor. The key is then fed into the crypto module. The crypto module in this employs a simple encoding process, which is essentially the process of permutation of the bits in the packet according to the dynamically created permutation code generated via RC4. The encoding is a simple encryption mechanism adopted for cryptography. Such architecture allows for adoption of stronger encryption mechanisms in line of encoding. Lastly, the forwarding module handles the process of sending or receiving of encoded packets along the path to the sink.

D. Operational Modes

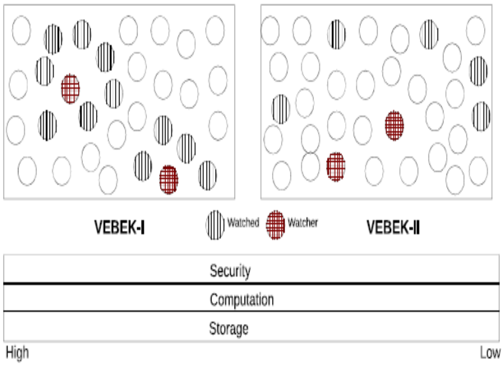


Fig. 3. Operational modes

The protocol provides three security services: Authentication, integrity, and non-repudiation. The fundamental notion behind providing these services is the watching mechanism described before. The watching mechanism requires nodes to store one or more records (i.e., current virtual energy level, virtual bridge energy values, and Node-Id) to be able to compute the dynamic keys used by the source sensor nodes, to decode packets, and to catch erroneous packets either due to communication problems or potential attacks. However, there are costs (communication, computation, and storage) associated with providing these services. In reality, applications may have different security requirements. For instance, the security need of a military WSN application (e.g., surveiling a portion of a combat zone) may be higher than that of a civilian application (e.g., collecting temperature data from a national park). The VEBEK framework also considers this need for flexibility and thus, supports two operational modes: VEBEK-I and VEBEK-II. The operational mode of VEBEK determines the number of nodes a particular sensor node must watch. Depending on the vigilance required inside the network, either of the operational modes can be configured for WSN applications. Different modes and the range of associated costs of each mode are given in Figureure3. The details of both operational modes are given below.

1) VEBEK-I

In the VEBEK-I operational mode, all nodes watch their neighbors; whenever a packet is received from a neighbor sensor node, it is decoded and its authenticity and integrity are verified. Only legitimate packets are forwarded toward the sink. In this mode, we assume there exists a short window of time at initial deployment that an adversary is not able to compromise the network, because it takes time for an attacker to capture a node or get keys. During this period, route initialization information may be used by each node to decide which node to watch and a record r is stored for each of its 1-hop neighbors in its watch-list. To obtain a neighbor's initial energy value, a network-wise master key can be used to transmit this value during this period similar to the shared-key discovery phase of other dynamic key management schemes. Alternatively, sensors can be pre-loaded with the initial energy value. When an event occurs and a report is generated, it is encoded as a function of a dynamic key based on the virtual energy of the originating node, and transmitted. When the packet arrives at the next-hop node, the forwarding node extracts the key of the sending node (this could be the originating node or another forwarding node) from its record (the virtual perceived energy value associated with the sending node and decodes the packet). After the packet is decoded successfully, the plaintext ID is compared with the decoded ID. In this process, if the forwarding node is not able to extract the key successfully, it will decrement the predefined virtual energy value from the current perceived energy and tries another key before classifying the packet as malicious (because packet drops may have occurred due to communication errors). This process is repeated several times; however, the total number of trials that are needed to classify a packet as malicious is actually governed by the value of VirtualKeySearchThreshold. If the packet is authentic, and this hop is not the final hop, the packet is re-encoded by the forwarding node with its own key derived from its current virtual bridge energy level. If the packet is illegitimate, the packet is discarded. This process continues until the packet reaches the sink. Accordingly, illegitimate traffic is filtered before it enters the network.

Re-encoding at every hop refreshes the strength of the encoding. Recall that the general packet structure is [ID, {ID, type, data} k]. To accommodate this scheme, the ID will always be the ID of the current node and the key is derived from the current node's local virtual bridge energy value. If the location of the originating node that generated the report is desired, the packet structure can be modified to retain the ID of the originating node and the ID of the forwarding node.

VEBEK-I reduces the transmission overhead as it will be able to catch malicious packets in the next hop, but increases processing overhead because of the decode/encode that occurs at each hop.

2) VEBEK-II

In the VEBEK-II operational mode, nodes in the network are configured to only watch some of the nodes in the network. Each node randomly picks r nodes to monitor and stores the corresponding state before deployment. As a packet leaves the source node (originating node or forwarding node) it passes through node(s) that watch it probabilistically. Thus, VEBEK-II is a statistical filtering approach like SEF [2] and DEF [3]. If the current node is not watching the node that generated the packet, the packet is forwarded. If the node that generated the packet is being watched by the current node, the packet is decoded and the plaintext ID is compared with the decoded ID. Similar to VEBEK-I, if the watcher-forwarder node cannot find the key successfully, it will try as many keys as the value of VirtualKeySearchThreshold before actually classifying the packet as malicious. If the packet is authentic, and this hop is not the final destination, the original packet is forwarded unless the node is currently bridging the network. In the bridging case, the original packet is re-encoded with the virtual bridge energy and forwarded. Since this node is bridging the network, both virtual and perceived energy values are decremented accordingly. If the packet is illegitimate, which is classified as such after exhausting all the virtual perceived energy values within the VirtualKeySearchThreshold window, the packet is discarded. This process continues until the packet reaches the sink.

This operational mode has more transmission overhead because packets from a malicious node may or may not be caught by a watcher node and they may reach the sink (where it is detected). However, in contrast to the VEBEK-I mode, it reduces the processing overhead (because less re-encoding is performed and decoding is not performed at every hop). The trade-off is that an illegitimate packet may traverse several hops before being dropped. The effectiveness of this scheme depends primarily on the value r, the number of nodes that each node watches. Note that in this scheme, re-encoding is not done at forwarding nodes unless they are bridging the network.

III. Analysis of Rekeying Cost for WSNs

One significant aspect of confidentiality research in WSNs entails designing efficient key management schemes. This is because regardless of the encryption mechanism chosen for WSNs, the keys must be made available to the communicating nodes (e.g., sources, sink(s)). The keys could be distributed to the sensors before the network deployment or they could be re-distributed (rekeying) to nodes on demand as triggered by keying events. The former is static key [13] management and the latter is dynamic key [14] management. There are myriads of variations of these basic schemes in the literature. In this chapter, we only consider dynamic keying mechanisms in our analysis since VEBEK uses the dynamic keying paradigm. Dynamic keying schemes go through the phase of rekeying either periodically or on demand as needed by the network to refresh the security of the system. With rekeying, the sensors dynamically exchange keys that are used for securing the communication. Hence, the energy cost function for the keying process from a source sensor to the sink while sending a message on a particular path with dynamic key-based schemes can be written as follows (assuming computation cost, E_{comp}, would approximately be fixed):

$$E_{Dyn} = (E_{Kdisc} + E_{comp}) * E[\eta_h] * \chi / \tau \tag{1}$$

where χ is the number of packets in a message, τ is the key refresh rate in packets per key, E_{Kdisc} is the cost of shared-key discovery with the next hop sensor after initial deployment, and $E[\eta_h]$ is the expected number of hops. In dynamic key-based schemes, r may change periodically, on-demand, or after a node-compromise. A good analytical lower bound for $E[\eta_h]$ is given in [15] as

$$E[\eta_h] = (D-t_r) / (E[d_h]) + 1 \tag{2}$$

where D is the end-to-end distance (m) between the sink and the source sensor node, tr is the approximated transmission range (m), and $E[d_h]$ is the expected hop distance (m) [16]. An accurate estimation of $E[d_h]$ can be found in [16]. Finally, E_{Kdisc}, can be written as follows:

$$E_{Kdisc} = \{(E\,[N_e] + 1) * E_{node} * M - E[N_e] * \{E_{tx} + E_{rx})\} \tag{3}$$

$$E_{node} = E_{tx} + E_{rx+} E_{comp} \tag{4}$$

where E_{node} is the approximate cost per node for key generation and transmission, $E[N_e]$ is the expected number of neighbors for a given sensor, M is the number of key establishment messages between two nodes, and E_{tx} and E_{rx} are the energy cost of transmission and reception, respectively. Given the transmission range of sensors (assuming bi-directional communication links for simplicity), t_r, total deployment area, A, total number of sensors deployed, N, $E[N_e]$ can be computed as

$$E\,[N_e] = (N * \prod * t_r^2)/A \tag{5}$$

IV. Notations

The notations used throughout this paper are represented in the below table Table1.

TABLE I: Notions Used

E_{tx}	Tx energy	E_{sens}	Sensing energy
E_{rx}	R_x energy	E_{sa}	Staying alive energy
E_{comp}	Computation energy	E_{vc}	Virtual cost
E_{enc}	Encoding energy	E_p	Perceived energy
E_{dec}	Decoding energy	E_b	Bridge energy
E_{fw}	Forwarding energy	E_{Kdisc}	Key discovery energy
E_{Dyn}	Dynamic energy	E_{so}	Source node energy
$E[\eta_h]$	Expected # of hops	P_{drop}	Drop probability
ψ	Sync ratio	L	Packet size
N	# of nodes	R	# of watch nodes

V. Design

1) Use case Diagram

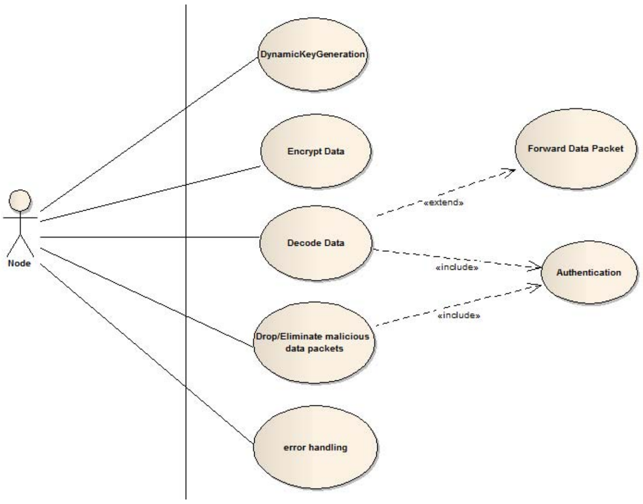


Fig. 4. Use case diagram

2) Activity Diagram

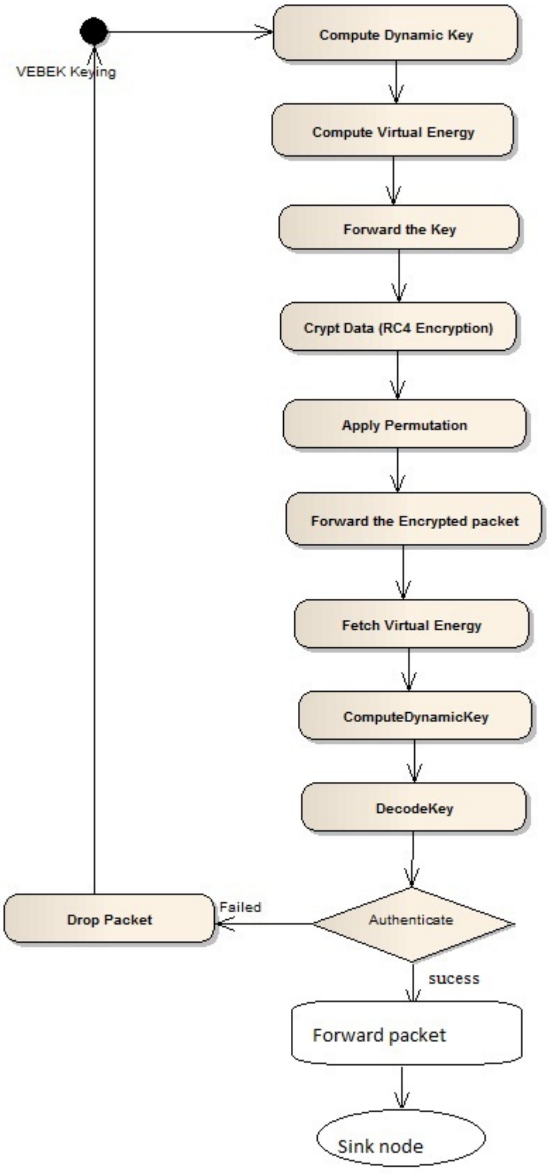


Fig. 5. Activity diagram

An activity diagram shows the flows from activity to activity within a system. They address the dynamic view of a system.

3) Topology

The initial designed topology is shown below. The topology contains 25 sensor nodes in the network, with one base station node to which all the nodes send the senor sensed information through their cluster heads.

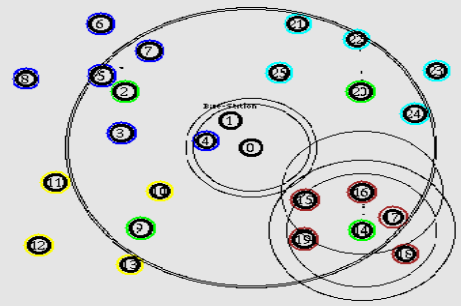


Fig. 6. Topolgy of WSN

An unauthorized user or node is also identified and marked as the attacker and the information of that node is transmitted to the remaining nodes in the network so that the node is blocked and even if the packets are transmitted by that node the packets are dropped. The identification of the attacker is depicted in the below Figure 7.4. In this topology node 14 is identified as the attacker even at the later stages some other nodes like node18 are also identified

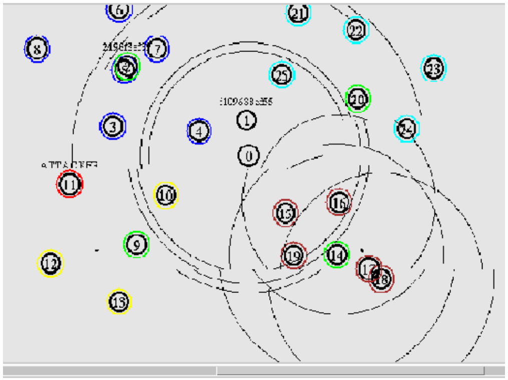
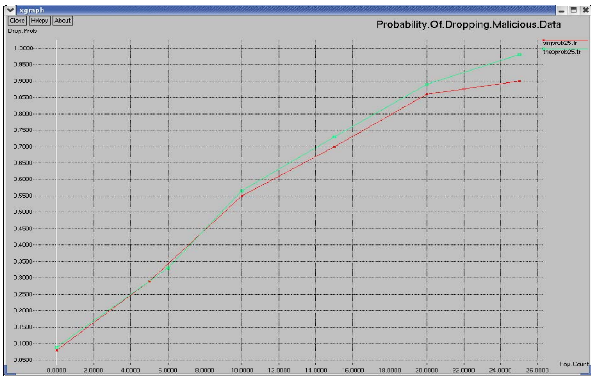


Fig. 7. Identification of attacker with red circle

VI. Comparisions

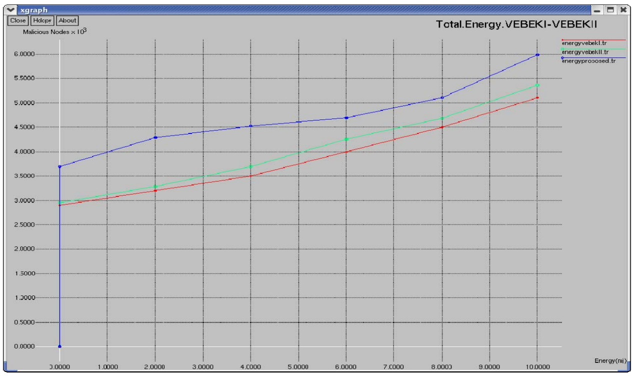
In both the operational modes there is a single cost (E_{So}) to stay-alive, sense the event, encode the packet, and transmit the packet (E_{sa}, E_{sens}, E_{enc}, E_{tx}) at the source sensor. Thus,

$$E_{So}=E_{sens}+E_{enc}+E_{tx}+E_{sa}$$



Graph1: Probability of dropping malicious packets

The total energy consumptions by this protocol is calculated and depicted in the below graph.



Graph2: Total Energy consumptions

VII. Conclusion

In this paper, efficient and secure communication frameworks have been developed for WSN applications. Motivated by the downsides of current dynamic key management and en-route-filtering schemes, the fact that the communication cost is the most dominant factor in a sensor's energy consumption [10], and further building upon the concept of sharing a dynamic cryptic credentials, security to sensor-based applications was addressed using a new approach. As opposed to other "chatty" dynamic key management and en-route filtering schemes, the focus was on eliminating specific control messages for keying or rekeying in the network so that some of the energy savings from transmission cost could be utilized for the computation of local security operations.

As emphasized multiple times previously within the context of this thesis, communication is very costly for WSNs and for certain WSN applications. Independent of the goal of saving energy, it may be very important to minimize the exchange of messages (e.g., military scenarios). A secure communication framework for WSNs was developed based on the idea of sharing a dynamic cryptic credential and the residual virtual energy of the sensor was used intelligently as a dynamic cryptic credential.

Acknowledgment

Paper completion fetches any individual fair amount of happiness, which cannot be expressed in words. Likewise I felt the same way after successful completion of my work that happened only because of auxiliary cooperation from some people.

I take it as a great privilege to express our heartfelt gratitude to the Principal of the college Dr. G. Ranga Janardhana and Head of the department Dr. MHM Krishna Prasad and Sri O. Srinivasa Rao, HCSE of JNTUK – Vizianagaram Campus for his precious guidance and constant encouragement.

I am thankful to all faculty members for extending their kind cooperation and assistance. Finally, I am extremely thankful to my parents and friends for their constant help and moral support.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey,” *Computer Networks*, vol. 38, no. 4, pp. 393-422, Mar. 2002.

[2] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," *IEEE JSAC*, vol. 23, no. 4, pp. 839-850, April 2005.

[3] Z. Yu and Y. Guan, "A dynamic en-route scheme for filtering false data injection in wireless sensor networks," in *Proceeding of IEEE INFOCOM*, pp. 1-12, April 2006.

[4] C. Kraub, M. Schneider, K. Bayarou, and C. Eckert, "Stef: A secure ticket-based en-route filtering scheme for wireless sensor networks," *The 2nd Int. Conf. on Availability, Reliability and Security (ARES)*, pp. 310-317, April 2007.

[5] M. Ma, "Resilience of sink filtering scheme in wireless sensor networks," *Elsevier Comput. Commun.*, vol. 30, no. 1, pp. 55-65, 2006.

[6] J. Hyun and S. Kim, "Low energy consumption security method for protecting information of wireless sensor networks," *LNCS Advanced Web and Network Technologies, and Applications*, vol. 3842, pp. 397-404, 2006.

[7] S. Zhu, S. Setia, S. Jajodia and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proc. of IEEE Symposium on Security and Privacy*, 2004.

[8] H. Hou, C. Corbett, Y. Li and R. Beyah, "Dynamic energy-based encoding and filtering in sensor networks," in *Proc. of the IEEE MILCOM*, October 2007.

[9] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. of ACM MOBICOM*, August 2002, pp. 56-67.

[10] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. Tygar, "Spins: Security protocols for sensor networks," in *Proc. of MOBICOM'01*, 2001.

[11] M. Luk, G. Mezzour, A. Perrig and V. Gligor, "Minisec: A secure sensor network communication architecture," *6th International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 479-488, April 2007.

[12] R. Roman, C. Alcaraz, and J. Lopez, “A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes,” *Mobile Networks and Applications*, Springer, vol. 12, no. 4, pp. 231-244, August 2007.

[13] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *Proc. of IEEE Symposium on Security and Privacy*, 2002, pp. 41-47.

[14] M. Eltoweissy, M. Moharrum, and R. Mukkamala, “Dynamic key management in sensor networks,” *IEEE Communications Magazine*, vol. 44, no. 4, pp. 122-130, April 2006.

[15] M. Vuran and I. Akyildiz, “Cross-layer analysis of error control in wireless sensor networks,” in *Proceeding of IEEE SECON*, vol. 2, pp. 585-594, Sept. 2006.

[16] M. Zorzi and R. Rao, "Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance," *Mobile Computing, IEEE Transactions on*, vol. 2, no. 4, pp. 337-348, Oct.-Dec. 2003.

K. Naga Krishnaja received the honors of bachelor degree in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad, India in May 2006 and received Masters degree in Information and Technology from Jawaharlal Nehru Technology University from Kakinada, Andhra Pradesh, India with distinction and further continuing her research work in the field of wireless sensor networks and planning for the PhD.

She has a good experience at the content development and proof reading of the publication books for bachelors and master degree courses. She also published a paper in IPCSIT International Conference on Advancements in Information Technology in 2011.