International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Novel Approach of Node Clone Detection in Wireless Sensor Networks

G. David[1], K. Srujana[2]

[1]PG Student, Department of CSE Prakasm Engineering College, Kandukur, Andhra Pradesh, India

[2]Associate Professor, CSE Department, Prakasam Engineering College, Kandukur, Prakasam (Dt), India

Abstract: ***Wireless Sensor Networks (WSNs) are often deployed in hostile environments where an adversary can physically capture some of the nodes, first can reprogram, and then, can replicate them in a large number of clones, easily taking control over the network. A wireless sensor network is a collection of nodes organized in to a cooperative network. This network is prone to various attacks due to poor security .A few distributed solutions to address this fundamental problem have been recently proposed.Wireless sensor networks are vulnerable to the node clone, and several distributed protocols have been proposed to detect this attack. However, they require too strong assumptions to be practical for large-scale, randomly deployed sensor networks. In this paper, we propose two novel node clone detection protocols with different tradeoffs on network conditions and performance. The first one is based on a distributed hash table (DHT), by which a fully decentralized, key-based caching and checking system is constructed to catch cloned nodes effectively. The protocol performance on efficient storage consumption and high security level is theoretically deducted through a probability model, and the resulting equations, with necessary adjustments for real application, are supported by the simulations. Although the DHT-based protocol incurs similar communication cost as previous approaches, it may be considered a little high for some scenarios. To address this concern, our second distributed detection protocol, Our second distributed detection protocol named randomly directed exploration, presents good communication performance for dense sensor networks by a probabilistic directed forwarding technique along with random initial direction and border determination. The simulation results uphold the protocol design and show its efficiency on communication overhead and satisfactory detection probability.***

Keywords:

1. Introduction

Wireless sensor network, a network of sensor nodes, which are tiny with limited resources that communicate with each other to achieve a goal, through the wireless channels. This network is mainly used in military applications for monitoring security and in civil applications. This network is deployed in harsh and hostile environments. Based on operating nature, it is unattended and prone to various attacks. The basic security requirements of wireless sensor network are integrity, availability, confidentiality and communication. In this paper, we present two novel, practical node clone detection protocols with different tradeoffs on network conditions and performance. The first proposal is based on a distributed hash table (DHT) by which a fully decentralized, key based caching and checking System is constructed to catch cloned nodes. The Protocol is efficient in storage consumption and high level security. Our second protocol, named Randomly directed exploration, is intended to provide highly efficient communication.

1.1 Detection Protocols

Based on the detection methodologies, we classify two novel node clone detection protocols.

1. Distributed hash table (DHT)
2. Randomly directed exploration (RDE)

A. Distributed hash table (DHT)
Distributed hash table (DHT), by which a fully decentralized, key-based caching and checking system is constructed to catch cloned nodes. The protocol's performance on memory consumption and a critical security metric are theoretically deducted through a probability model, and the resulting equations, with necessary adjustment for real application, are supported by the simulations. In accordance with our analysis, the comprehensive simulation results show that the DHT -based protocol can detect node clone with high security level and holds strong resistance against adversary's attacks.

B. Randomly directed exploration (RDE)
This is intended to provide highly efficient communication performance with adequate detection probability for dense sensor networks. In the protocol, initially nodes send claiming messages containing a neighbor-list along with a maximum hop limit to randomly selected neighbors; then, the subsequent message transmission is regulated by a Probabilistic directed technique to approximately maintain a line property through the network as well as to incur sufficient randomness for better Performance on communication and resilience against adversary. In addition, border determination mechanism is employed to further reduce communication payload. During forwarding, intermediate nodes explore claiming messages for node clone detection. By design, this protocol consumes almost minimal memory, and the simulations show that it outperforms all other detection protocols in terms of communication cost, while the detection probability is satisfactory.

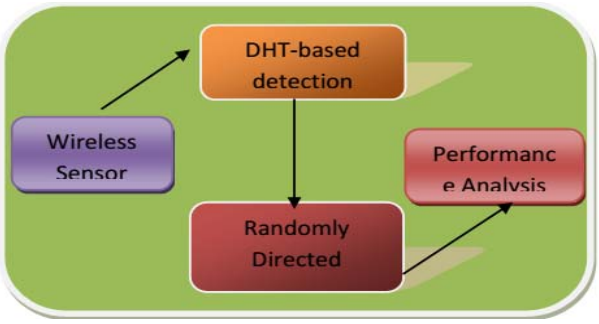

Figure 1: Block diagram

1.2 Existing System

WIRELESS sensor networks (WSNs) have gained a great deal of attention in the past decade due to their wide range of application areas and formidable design challenges. In general, wireless sensor networks consist of hundreds and thousands oflow-cost, resource-constrained, distributed sensors nodes, which usually scatter in the surveillance area randomly, working without attendance. If the operation environment is hostile, security mechanisms against adversaries should be taken into consideration. Among many physical attacks to sensor networks, the node clone is a serious and dangerous one. Because of production expense limitation, sensor nodes are generally short of tamper -resistance hardware components; thus, an adversary can capture a few nodes, extract code and all secret credentials, and use those materials to clone many nodes out of off-the-shelf sensor hardware. Those cloned nodes that seem legitimate can freely join the sensor network and then significantly enlarge the adversary's capacities to manipulate the network maliciously.

Disadvantages of Existing System

1. Among many physical attacks to sensor networks, the node clone is a serious and dangerous one.
2. Insufficient storage consumption performance in the existing system and low security level.

2. Network Model

We consider a WSN consisting of N wireless sensors, randomly deployed over a monitoring area. All sensors are assumed to be limited in communication and computation power as well as in battery life. On average, every sensor is able to directly communicate with d other sensors, referred to as *neighbors*.

Prior to deployment, every sensor is assigned a key pair (PK, SK). The key PK represents the node's public key. It is known to all the other network nodes and it is used as the node's unique identifier (ID). The key SK represents the node's secret key. We assume every sensor is able to determine its position using secure localization mechanisms [9].

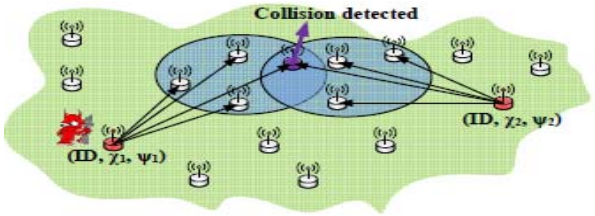

Figure 2: Graphical representation of the witness-based clone detection methods: a captured node and its clone are placed at different locations but possess the same ID. Periodically, the captured node and the clone choose their set of nodes to send a message, containing their ID and the current location (_i, i). If the intersection of the two chosen sets is non-empty, then a collision in received messages occurs and a clone is detected.

a) Adversarial Model

We consider a time-persistent adversary, e.g., an adversary who operates over an extended period of time. The adversary compromises a set of network nodes and extracts their information, such as sensed data, the states of the network protocols and the assigned cryptographic secrets. Using the extracted data, the adversary then fabricates exact functional copies of captured nodes (*clones*) and deploys the clones back into the network. Every captured node is assumed to be cloned at least once.

b) Detection of Clone Attacks

For sensor u, its fingerprint is computed from the code words collected from its neighborhood $N(u)$. As stated in Section 3, sensors are stationary after deployment. A legitimate sensor u belongs to a "fixed" neighborhood, whose social characteristics can be encoded into u's fingerprint. Therefore, each sensor is required to "sign" with its fingerprint FPu whenever it generates a new message to send to the base station. The message transmission should be in the following format2: $u \rightarrow BS : \{IDu, FPu, content\}$
Assume X is the superimposed s-disjunction code to generated the social codeword for each sensor, which can be represented by an $M \times N$ matrix. According to Algorithm 1, the length of a fingerprint is $\log2(M)$. Even with $M = 100,000$, a fingerprint takes no more than 2 bytes to be included in a message. Hence, our detection algorithm imposes a very slight message overhead for protecting a sensor network against clone attacks.

In our consideration, a cloned sensor may use an arbitrary fingerprint (e.g. the fingerprint of the original sensor), or compute a new fingerprint that is consistent with its new residency. Hence, detecting clone attacks should be conducted in two aspects:

2.1 Detection at the Sensor Side

Suppose u generates a new message, which is forwarded to a neighbor $v \in N(u)$. Sensor v should check whether the enclosed fingerprint FPu is consistent with the one in its record that v has computed for u previously. Once vi dentifies any mismatch, v should raise an alarm to the base station. Then, the base station should send a query to the neighborhood $N(u)$. Each sensor in $N(u)$ should reply toBS with its own record about FPu. Thus, BS can determine which sensor should be revoked afterwards3.The local

fingerprint verification ensures that no sensor can use a fingerprint that is not consistent with its neighborhood. Note that a legitimate sensor u derives its fingerprint based on the information retrieved from its neighborhood.

The local information exchange ensures that u's neighbors can also compute u's fingerprint independently. Thereafter, though a cloned sensor u can "pretend" to be legitimate by having all the valid security information, it cannot cheat its neighbors that can easily tell whether u is using an inconsistent fingerprint.

2.2 Detection at the Base Station

The base station should maintain a fingerprint file indexed with sensor IDs, and insert an entry for sensor u upon receiving 's first message. After BS receives a new message c from sensor u, it checks whether the fingerprint FPu in$C$ matches its record obtained from u's previous messages. If FPu does not match the record, there must be a clone attack in the network. Then, BS may broadcast a revoke message concerning sensor u throughout the network, such that those sensors with the ID u will be isolated afterwards.

The detection at the base station is to work against a "smart" clone that intelligently computes a new fingerprint consistent with its current neighborhood so as to escape from being identified by the neighboring sensors. By establishing a fingerprint file, the base station can easily determine whether there exist several sensors in the network that use different fingerprints but with the same ID.

3. Security Analysis

In this section, we analyze the impact of compromised and cloned nodes on our detection algorithm. We observe that an adversary can launch effective clone attacks at the following two scenarios. Note that a clone attack at other scenarios can be detected via the fingerprint computed by Algorithm 1 with a much less effort.

- Node compromise/clone during fingerprint generation,
- Node compromise/clone during the detection phase.
- Next, we will study the resilience of our scheme under these two scenarios, and quantify the effectiveness of our detection by studying the detection probability.

3.1 Node compromise/clone at fingerprint generation

Assume an adversary compromises a sensor u right after deployment, replicates and distributes the clones before the fingerprint generation is finished in the network. Then u's clones, say u_1, u_2, ..., u_t, can participate in the fingerprint generation procedure as a legitimate sensor. Since the clones are deployed into different locations, their derived fingerprints will be different. Thus, the base station can easily identify these clones, which have the same ID ($IDu = IDu\_1 = ... = IDu\_t$) but different fingerprints.

Note that there is no impact on a legitimate sensor w, if a clone node u_i is inserted into the neighborhood $N(w)$. Sensor w can safely use its fingerprint which may contain the codeword contributed by the clone u_i though, since the other legitimate neighbors in $N(w)$ will use the codeword of u_i as well. It does not affect the effectiveness of w's fingerprint against clone attacks, because w's fingerprint is based on the neighborhood difference and social relationships rather than an individual codeword.

3.2 Node compromise/clone at the detection phase

Assume an adversary compromises a sensor u, replicates and distributes the clones after all the legitimate sensors have derived their fingerprints. Then for a cloned sensor v, the adversary may determine v's fingerprint FPv according to the following methods:

- *Case I: Sensor v selects FPv = FPu.* Suppose sensor v generates a message C and forwards it to a neighbor $w \in N(v)$. If w is legitimate, w should raise an alarm since no match can be found in w's fingerprint records. Then the base station can identify the clone v after checking the fingerprints belonging to $N(v)$. Unless the adversary completely compromises and controls the neighborhood $N(v)$, the clone v will be identified by our detection scheme.

  Note that there is no incentive for the adversary to compromise all nodes in $N(v)$ in order to launch a clone attack. Furthermore, for a cloned area (containing cloned nodes only) that is larger than a typical open neighborhood, all the boundary nodes can be easily identified and then revoked. Thus, the whole compromised/ cloned region will be isolated. Our detection scheme is robust against colluding attackers.
- *Case II: Sensor v selects an arbitrary bit stream as FPv.* Same as *case I*.
- *Case III: Sensor v computes FPv based on the codeword's from its neighborhood* $N(v)$. A smart clone tries to escape from being identified by its neighbors, and computes a fingerprint consistent with its neighborhood.

  Assume the adversary is powerful enough to listen on the codeword's broadcasted around $N(v)$ and help v compute its FPv. However, after receiving messages from sensors u and v, the base station will find out that $IDu = IDv$ but $FPu \_= FPv$. Then the base station identifies the existence of a clone attack, and therefore revokes all the sensors with IDu.

3.3 Detection Probability

In the following, we investigate the probability *Pun detected* that a clone node escapes from being detected successfully. Assume the adversary compromises a sensor u, clones t copies of u (denoted as u_1, u_2, ..., u_t), and distributes the clones into the network. To avoid being detected, these t clones must fulfill the following two requirements simultaneously:

- *Condition I*: All the clones u_1, u_2, ..., u_t must use the same fingerprint as the sensor u. Otherwise, the base station will identify the difference among the fingerprints used by these nodes that share the same IDs.
- *Condition II*: Each of the clones u_1, u_2, ..., u_t must use a fingerprint that is consistent with its current neighborhood. Otherwise, the cloned node will be identified by their neighbors. Thereafter, only when the

neighbors of the *t* clones contribute to the same fingerprint as that of sensor *u*, our detection algorithm fail to identify.

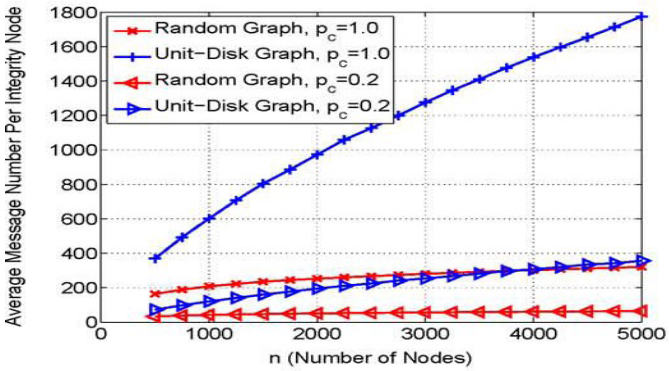

Figure 1: Simulation results of DHT detection on number of nodes

4. Randomly Directed Exploration

The problems associated with the dht are it incurs more communication cost because of the chord overlay network and thus it is sensitive to energy and storage consumption. To overcome these problems a new node clone detection protocol introduced namely randomly directed exploration. Here the each node only needs to know and buffer a neighbor list having all neighbors ID and locations. During detection round each node constructs claiming message with signed version of neighbor list and then deliver message to others which will compares with its own neighbor list to detect node clone. If there exists any node clone, one witness node successfully catches the clone and notifies the entire network by broadcasting. The efficient way to achieve randomly directed exploration needs some mechanisms and routing protocols. First the claiming message needs to provide maximum hop limit and it is sent to random neighbors. Then the further message transmission will maintain a line and this transmission line property enables a message to go through a network as fast as possible[6]. The communication cost of this protocol is low and it is limited by the border determination mechanism. And the assumption made here is that each node knows about its neighbors locations.

Detection round:
Initially the node clone detection round is activated by the initiator. At the right mentioned action time, each node creates its own neighbor list (ID of neighbor and location). Then that node act as an observer for all its neighbors and starts to generate claiming messages. The claiming message involves node ID, location and its neighbor list[6]. The claiming message by node is constructed by Mα=ttl, idα, Lα, neighbor list where ttl is time to live.

Algorithm 1: rde-processmessage Mα: An intermediate node processes a message
1: verify the signature of Mα
2: compare its own neighbor-list with the neighbor-list in Mα
3: **if** found clone **then**
4: broadcast the evidence;
5: ttl<=ttl-1
6: **if** ttl ≤ **0 then**
7: discard Mα
8: **else**
9: next node<=get next node (Mα) {See Algorithm 4}
10: **if** next node =NIL **then**
11: discard Mα
12: **else**
13: forward Mα to next node[6]

The intermediated nodes will change the value of ttl during transmission. In each time, the node transmits message to a random neighbor. When an intermediate node β receives a claiming message Mα, it launches rde-process message Mα. During the processing the node clone is detected by comparing the neighbor list of node which acts as inspector β with neighbor list in the message. If clone detected then the witness node β will broadcast an evidence message M evidence= (Mα,Mβ) to notify the whole network such that the cloned nodes are removed from the network[6]. Node decreases the message's ttl by 1 and discards the message if ttl reaches zero during routing; otherwise it will query Algorithm 4 to determine the next node receiving the message.

Algorithm 2: get next node (Mα): To determine the next node that receives the message
1: determine ideal angle, target zone, and priority zone
2: **if** no neighbors within the target zone **then**
3: **return** NIL
4: **if** no neighbors within the priority zone **then**
5: next node<= the node closest to ideal angle
6: **else**
7: next node<= a probabilistic node in the priority zone, with respect to its probability proportional to angle distance from priority
zone border
8: **return** next node[6].

A. Deterministic directed transmission: The ideal direction can be calculated when node receives a claiming message from previous node and the next destination node should be closest to the ideal direction for the best effect of line transmission.
Network border determination: The communication cost is reduced by taking network shape into consideration. Due to physical constrains in many sensor network applications, there exist outside borders. The claiming message can be directly discarded when reaching some border in the network. To determine a target zone then no neighbor is found in this zone, target range is used along with ideal direction, the current node will conclude that the message has reached a border, and thus throw it away.

B. Probabilistic directed transmission: priority range along with the ideal direction is used to specify a priority zone, in which the next node will be selected. The deterministic directed candidate within the target zone will be selected as the next node when no nodes are located in that zone,. If there are several nodes in the priority zone, their selection probabilities are proportional to their angle distances to priority zone border. As a result, to reduce detection probability dramatically the adversary may remove some nodes in strategic locations Claiming messages transmissions from a cloned node's neighbors are highly correlated, which affects the protocol communication and

security performance[1]. Those drawbacks are overcome, by the probabilistic directed mechanism, and the protocol performance is improved significantly

C. Performance Analysis of RDE

Communication cost: The RDE's communication cost depends on the routing parameter settings. On average, there are r claiming messages sent by each observer, and each message transmits at most ttl hops , r is a constant small number, say 1 for a dense network, but ttl is generally related to the network size . So ttl=√n because there are nodes in the network, and by the line property of protocol routing, it is very likely for any two nodes to be reachable within √n hops for a normal network topology[6]. In other words, ttl=√n would be sufficient for messages to go across the network. The upper bound of communication cost in the randomly directed exploration protocol is O√n and its shown in fig 4 (a).

D. Detection probability: Relieving message-discarding and protecting witness are achieved by random initial direction and probabilistic directed transmission. By them, there is no critical location to affect message transmission, which limits the capacity of message-discarding, and every neighbor of a cloned node has similar potential to become witness so it is hard for the adversary to get rid of witness in advance[1]. The RDE protocol's detection probability is determined by the number of nodes that are reached when randomly drawing lines where each has a random initial angular and fixed number of nodes along this direction with the border limitation. Let h denote the reachable node number; θ, it is a function of (an initial angular),ttl (the number of maximum hops), and v (the number of the claiming messages). Therefore, for a network with n nodes, the detection probability is given by P RDE=h (ttl,θ,v)/n shown in fig 4(b).

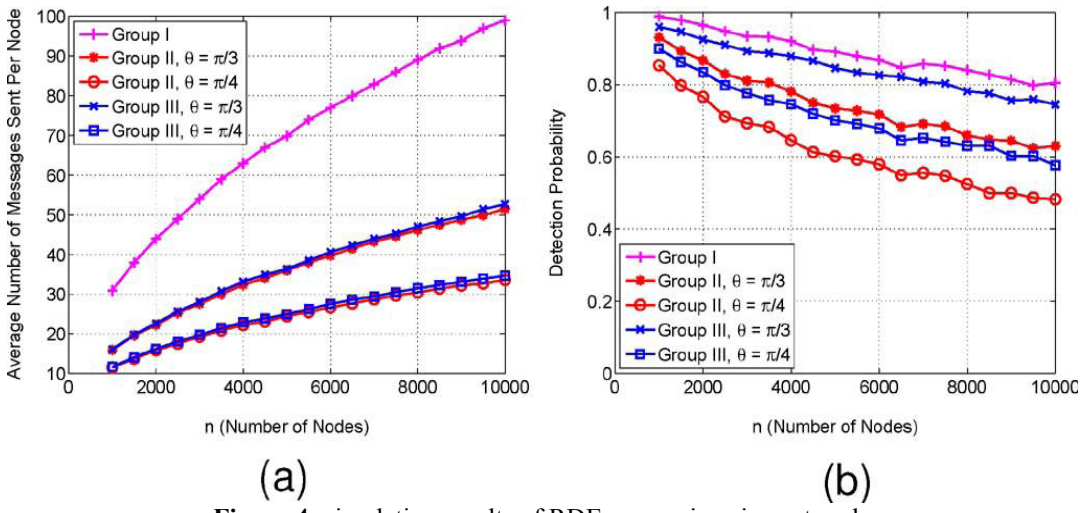

Figure 4: simulation results of RDE on varying size network.

E. Storage consumption: The RDE protocol is exceedingly memory-efficient. It does not rely on broadcasting; thus, no additional memory is required to suppress broadcasting flood. The protocol does not demand intermediate nodes to buffer claiming messages, all memory requirement lies on the neighbor-list, which, in fact, is a necessary component for all distributed detection approaches. Therefore, the protocol consumes almost minimum memory.

5. Conclusion

Sensor nodes lack tamper-resistant hardware and are subject to the node clone attack. So two distributed detection protocols are presented: One is based on a distributed hash table, which forms a Chord overlay network and provides the key-based routing, caching, and checking facilities for clone detection, and the other uses probabilistic directed technique to achieve efficient communication overhead for satisfactory detection probability. While the DHT-based protocol provides high security level for all kinds of sensor networks by one deterministic witness and additional memory-efficient, probabilistic witnesses, the randomly directed exploration presents outstanding communication performance and minimal storage consumption for dense sensor networks.

From the analysis and simulation results, the randomly directed exploration protocol outperforms all other distributed detection protocols in terms of communication cost and storage requirements, while its detection probability is satisfactory, higher than that of line-selected multicast scheme.

References

[1] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proc. IEEE Symp. Security Privacy*, 2005, pp. 49–63.

[2] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up data in P2P systems," *Commun. ACM*, vol. 46, no. 2, pp. 43–48, 2003.

[3] Y. Zhang,W. Liu,W. Lou, andY. Fang, "Location-based compromise tolerant security mechanisms for wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 247–260, Feb. 2006.

[4] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proc. 10th ACM CCS*, Washington, DC, 2003, pp. 62–72.

[5] R. Anderson, H. Chan, and A. Perrig, "Key infection: Smart trust for smart dust," in *Proc. 12th IEEE ICNP*, 2004, pp. 206–215.
[6] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei, "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," in *Proc. 8thACMMobiHoc*,Montreal, QC, Canada, 2007, pp. 80–89.
[7] B. Zhu, V. G. K. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient distributed detection of node replication attacks in sensor networks," in *Proc. 23rd ACSAC*, 2007, pp. 257–267.
[8] H. Choi, S. Zhu, and T. F. La Porta, "SET: Detecting node clones in sensor networks," in *Proc. 3rd SecureComm*, 2007, pp. 341–350.
[9] R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir, "On the detection of clones in sensor networks using random key predistribution," *IEEE Trans. Syst.s, Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1246–1258, Nov. 2007.
[10] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACM Conf. Comput. Commun. Security*, Washington, DC, 2002, pp. 41–47.
[11] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. CRYPTO*, 1984, LNCS 196, pp. 47–53.
[12] R. Poovendran, C. Wang, and S. Roy, *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*. New York: Springer-Verlag, 2007.
[13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.
[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," in *Proc. SIGCOMM*, San Diego, CA, 2001, pp. 161–172.
[15] I. Stoica, R.Morris, D. Liben-Nowell, D. R. Karger,M. F.Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, Feb. 2003.
[16] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proc. IFIP/ACM Int. Conf. Distrib. Syst. Platforms Heidelberg*, 2001, pp. 329–350.
[17] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. 1st Int. Conf. Simulation Tools Tech. Commun., Netw. Syst. Workshops*, Marseille, France, 2008, pp. 1–10.
[18] A. Awad, C. Sommer, R. German, and F. Dressler, "Virtual cord protocol (VCP): A flexible DHT-like routing service for sensor networks," in *Proc. 5th IEEE MASS*, 2008, pp. 133–142.
[19] R. Diestel, *Graph Theory*, 3rd ed. New York: Springer, 2006.
[20] H. Chan and A. Perrig, "Security and privacy in sensor networks," *Computer*, vol. 36, no. 10, pp. 103–105, 2003.
[21] X. Wu, G. Chen, and S. K. Das, "On the energy hole problem of nonuniform node distribution in wireless sensor networks," in *the Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems*, 2006, pp180–187.

Author Profile

MR G.DAVID received B. Tech. degree in Computer Science and Information technology from JNTUK University, in 2012 Currently he is doing M. Tech. in Prakasam Engineering College, from JNTUK University, Kakinada, India.

K. Srujana, received B_level M.Tech completed in Sathya Bhama University in 2008, and currently she is working as an Associate Professor, Department of CSE in Prakasam Engineering College, Kandukur, India

.