ISSN: 2321-7782 (Online)
Volume 2, Issue 11, November 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Hybrid Security Mechanisms for Intrusion Detection in Heterogeneous Wireless Sensor Network

B.Srinivasa Rao[1]
M.Tech scholar
Sri Sai Madhavi Institute of Science & Technology
Rajanagaram, Rajahmundry
India

Yuvaraju Chinnam[2]
Assoc.Professor, M.Tech (P.hd)
Sri Sai Madhavi Institute of Science & Technology
Rajanagaram, Rajahmundry
India

Abstract: ***Intrusion detection in Wireless Sensor Network (WSN) is of practical interest in many applications such as detecting an intruder in a battlefield. The intrusion detection is defined as a mechanism for a WSN to detect the existence of inappropriate, incorrect, or anomalous moving attackers. For this purpose, it is a fundamental issue to characterize the WSN parameters such as node density and sensing range in terms of a desirable detection probability. In this paper, we consider this issue according to two WSN models: homogeneous and heterogeneous WSN with security considerations along with pair wise key management with dispersive routes.***

Keywords: ***Homogeneous, Heterogeneous Sensor Networks, probabilistic key management, probabilistic authentication, hybrid network security***

I. INTRODUCTION

Detecting and tracking moving objects is a major class of applications in Sensor Networks, such as vehicle detection in military surveillance [1] and wild animal habitat monitoring [2]. These applications, by their nature, enforce certain detection quality and lifetime requirements. The first requirement determines how fast a Sensor Network should detect the intrusion of a moving vehicle, or how often the data about a wild animal should be sampled and collected. The second requirement specifies the working duration a Sensor Network should sustain. These two requirements, however, are conflicting optimization goals due to the stringent energy constraints of Sensor nodes.

Full sensing coverage is mandatory for Sensor monitoring applications that require either immediate response to detected events or information of all points in the sensing field. Full sensing coverage, however, is too expensive to support long-duration monitoring applications. More often those applications do not need zero response time or information at all points of the sensing field. Full sensing coverage provides over-qualified detection quality for these applications at the cost of exhausting network energy rapidly, which may be willing to sacrifice event detection probability or detection delay to some extent for increasing the network lifetime. A relaxed sensing coverage—partial coverage, where the sensing field is partially sensed by active sensors at any time—is a more appropriate approach to balancing object detection quality and battery power consumption.

A large number of practical sensing and actuating applications require immediate notification of rare but urgent events and also fast delivery of time sensitive actuation commands. In this paper, we consider the design of efficient wakeup scheduling schemes for energy constrained Sensor nodes that adhere to the bidirectional end-to-end delay constraints posed by such applications.

We evaluate several existing scheduling schemes and propose novel scheduling methods that outperform existing ones. We also present a new family of wakeup methods, called multi-parent schemes, which take a cross-layer approach where multiple

routes for transfer of messages and wakeup schedules for various nodes are crafted in synergy to increase longevity while reducing message delivery latencies.

Detection quality requirements are classified into average case detection quality requirements and worst-case detection quality requirements. The average-case detection quality can be characterized by the probability that a moving object is detected in a given observation duration, and by the average distance an object travels before detection. In contrast, the worst-case detection quality can be characterized by the lower limit of time duration to detect moving objects, and by the upper limit of distances that objects travel before detection..

II. Related Work

The preceding work in the field of key management for ireless Sensor Networks an be broken into two categories: schemes that have constant access to a KDC or rusted third-party keying mechanism and those that never do. While there has also been a large body of work on distributed certificate authorities applied to ad hoc networks, these schemes rely on public key cryptography. Emerging work in elliptic curve cryptography [3], [4] may one day allow such schemes to be deployed; however, symmetric key-based schemes are currently the most practical option. The wired world of networking is already familiar with a number of protocols for authentication and session key establishment. The classic protocol for authenticating communications between two machines was written by Needham and Schroeder [5]. Improvements to this protocol were made in the abundantly used Kerberos [6]. Fox and Gribble proposed the use of Charon [7], a proxy server designed to offload the memory overhead of Kerberos for Sensor devices. However, as these protocols require a large number of transmissions in order to establish a single session key with neighbors, they are therefore not particularly well suited for this environment.

A large number of key distribution schemes have been proposed for networks that are unable to access a KDC after deployment. The most famous of these predistribution schemes is the work by Eschenauer and Gligor [8]. Given a key pool of size P, nodes are preloaded with k keys (selected without replacement) such that two randomly picked nodes can communicate with a given probability (i.e., share at least one key). In order to determine whether or not a key is shared, each node broadcasts its key identifiers (which are randomly associated with the keys themselves before deployment) in plaintext. Neighbors sharing a key associated with one of those identifiers then issue a challenge/response to the source. If two nodes do not share keys directly, they can establish a session key with the help of neighbors with which a key is already shared. While this technique is well suited for establishing session keys in a stand-Vol 1 Issue 2, December 2012 132 alone network, it does not provide support for authentication.

Liu et al. [917] have modeled the intrusion detection problem in a Sensor WSN, where each Sensor is capable of moving. The authors have given the optimal strategy for fast detection and shown that Sensor WSN improves its detection quality due to the mobility of sensors.

A large number of sensing coverage maintenance protocols, aiming to conserve energy under various conditions, have been proposed ([10], [11], [12], [13], [14]). Yan et al. [14] presented an energy-efficient random reference point sensing protocol to achieve a targeted coverage degree. Nodes decide their active periods by exchanging reference points among neighbors. In [12], Hsin and Liu investigated coverage intensity and extensity of random sleep schedules and coordinated sleep schedules.

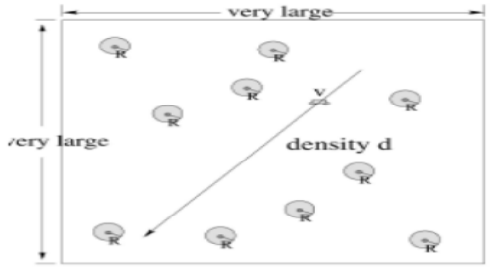


Fig 1: object detection and tracking scenario.

III. Proposed Approach

Many applications that make use of Sensor Networks require secure communication. Because asymmetric-key solutions are difficult to implement in such a resource constrained environment, symmetric-key methods coupled with a priori key distribution schemes have been proposed to achieve the goals of data secrecy and integrity. These approaches typically assume that all nodes are similar in terms of capabilities and, hence, deploy the same number of keys in all sensors in a network to provide the aforementioned protections. In this paper, we demonstrate that a probabilistic unbalanced distribution of keys throughout the network that leverages the existence of a small percentage of more capable Sensor nodes can not only provide an equal level of security, but also reduce the consequences of node compromise. To fully characterize the effects of the unbalanced key management system, we design, implement, and measure the performance of a complementary suite of key establishment protocols known as WIGER. Using their pre-deployed keys, nodes operating in isolation from external networks can securely and efficiently establish keys with each other.

IV. Network Model Intrusion Detection Model, Quality Metrics, and Application

We consider a WSN in a two-dimensional (2D) plane with N sensors, denoted by a set N={n1; n2; . . . ; nN}, where ni is the ith Sensor. These sensors are uniformly and independently deployed in a square area A=LxL. Such a random deployment results in a 2D Poisson point distribution of sensors All sensors are static once the WSN has been deployed. we consider two WSN types:

1. Homogeneous
2. Heterogeneous WSNs.

In a homogeneous WSN, each Sensor has the same sensing radius of rs, and the transmission range of rx. A sensor can only sense the intruder within its sensing coverage area that is a disk with radius rs centered at the Sensor. Denote the node density of the homogeneous WSN as (lambda). We then focus on a heterogeneous WSN with two types of Sensor.

1. Type I Sensor that has a larger sensing range rs1, as well as a longer transmission range rx1,
2. Type II Sensor that has a smaller sensing range rs2, as well as a shorter transmission range rx2.

The densities of Type I and Type II sensors are represented as lambda1 and lambda 2, respectively. WSN, where both Type I and Type II sensors follow the 2D Poisson point distribution. In a homogeneous or heterogeneous WSN, a point is said to be covered by a Sensor if it is located in the sensing range of any Sensor(s). The WSN is thus divided into two regions, the covered region, which is the union of all Sensor coverage disks, and the uncovered region, which is the complement of the covered region within the area of interest A. In our network model, the intruder does not know the sensing coverage map of the WSN

Fig. 1 shows a typical scenario of the object detection in a Sensor Network. There are a number of sensors deployed on a sensing field. A small object moves across the field along a randomly selected direction. The sensors perform their sensing tasks under some sensing schedules. In a sensing schedule, a node periodically wakes up and goes to sleep to conserve energy while meeting the object detection quality requirement.

» Sensors are randomly and independently distributed on the sensing field, with a density d.

» The Sensor Network is homogeneous, i.e., all sensors are identical. We denote the sensing range of a Sensor as R.

» The size of an moving object can be neglected, considering that it is significantly smaller than the sensing range of an individual Sensor.

» The object speed does not change during the detection process. We denote the object speed as v.

In reality, a moving object may change its moving speed and direction during detection, thus, it may be difficult for the surveillance center to obtain the precise speed of the object as well. Thus, given the range of object speed v based on past experiences, we can estimate the range of the corresponding object detection quality. Therefore, our model and analytical results are still useful for the cases of changing object speed.

In order to evaluate the average-case object detection quality of a Sensor Network, we define two metrics detailed as follows:

Detection Probability (DP): The detection probability is defined as the probability that an object is detected in a certain observation duration. *Average Stealth Distance (ASD):* The average stealth distance is defined as the average distance an object travels before it is detected for the first time.

For worst-case object detection quality of the network, we have the following two metrics:

Sufficient Phase (SP): The sufficient phase is defined as the smallest time duration in which an object is detected with 100 percent probability starting from any time for any position on the field where the object is initially located.

Worst-case Stealth Distance (WSD): The worst-case stealth distance is defined as the longest possible distance that an object travels before it is detected for the first time.

Taking energy constraints into account, we further define other two metrics.

Lifetime (LT): The system lifetime is the elapsed working time from system startup to the time when the object detection quality requirement cannot be met for the first time when live nodes continue sensing with their current periods.

Maximum Working Time (MWT): The maximum working time is the longest possible working time of the system that satisfies the object detection quality requirement. Contrary to the definition of the lifetime, in which nodes have fixed sensing periods, in the definition of the maximum working time, when some nodes deplete their power, the remaining nodes can adjust their sensing periods to sustain the object detection quality.

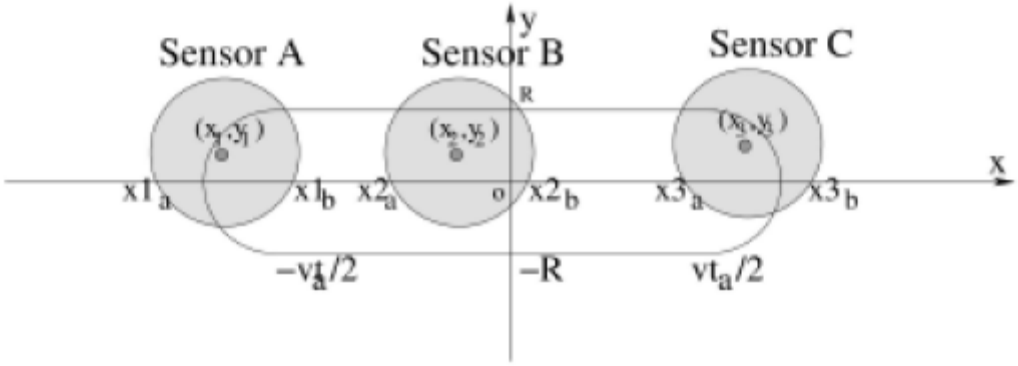


Fig. 2. Three sensors are located in the active area of a moving object.

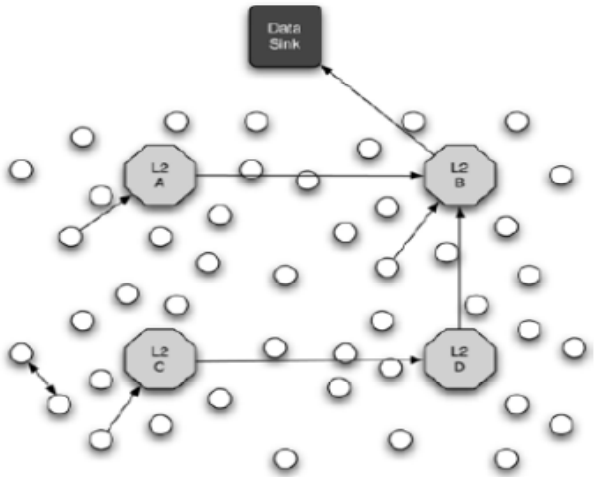
V. HETEROGENEOUS SENSOR NETWORKS

The previous work on random key pre-deployment in Sensor Networks has assumed either a grid or very large random-graph arrangement such that all neighbors within the transmission radius of a given node are reachable. Communication between adjacent nodes is therefore limited only by key matching. This model is not always realistic for a number of reasons. Primarily, it fails to take into account that signal-blocking barriers including hills, buildings, and walls may exist between neighboring nodes. More importantly, it fails to consider that better radio technology and resources are available to some

members of the network. Because the assumptions of topology and topography used in most previous approaches are violated in realistic settings, we propose a new network model.

Fig. 3 shows a network model with two main differences from that used in previous work. First, the landscape over which a Sensor Network is placed contains features that may segregate nodes into exclusive neighborhoods. Because nodes are still distributed through the same methods discussed in previous papers (e.g., dropped from an airplane), there is no way to determine a node's neighbors a priori. Due to the random nature of the deployment, the potential for node mobility and addition of nodes at a later time, assigning keys for specific neighbors is not possible. Second, instead of a homogeneous composition of nodes the network now consists of a mix of nodes with different capabilities and missions. The sensing or Level 1 (L1) nodes are assumed to be very limited in terms of memory and processing capability and perform the task of data collection

These nodes are indicated as white circles in Fig. 3. Level 2 (L2) nodes have more memory, processing ability, and additional radios (e.g., 802.11). These nodes are equipped with additional keys and take on the role of routers and gateways between networks. In addition to tamper-resistant casings, L2 nodes are assumed to be equipped with a fast encryption/deletion algorithm to protect their supplementary keys from compromise if they are captured.



VI. L2 Nodes Offer Access External Resources

Unbalanced Key Distribution

The session-key establishment phase attempts to create a secure communication link for pairs of nodes within a wireless transmission range that lack a shared key from the previous phase. If it is possible to communicate between a pair of nodes by using a multi-hop path through secure and trusted neighbors, then a key can be generated at the shared neighbor and distributed to the two endpoints. At the completion of this final phase, all nodes within transmission radius of each other should be able to communicate directly to whatever reliability the network designers have specified.

This basic scheme was extended by Chan et al. [4] by requiring nodes to share at least q keys with each other. In so doing, it becomes more difficult for an attacker to compromise communications as q instead of one key must first be captured. While providing robustness to compromise, the q-composite scheme has not previously been considered for the purpose of node authentication. Specifically, if a node can prove its possession of multiple keys known to be assigned to it, its identity can be probabilistically authenticated. Under the balanced scheme, an increase in q requires a significant growth in the number of keys stored by all L1 nodes. However, the unbalanced approach can reduce the burden of the q-composite scheme on L1 nodes while retaining its security advantages. The probability that two nodes containing key rings of differing sizes share exactly i keys is

$$p(i) = \frac{\binom{P}{i}\binom{P-i}{(m-i)+(k-i)}\binom{(m-i)+(k-i)}{m-i}}{\binom{P}{m}\binom{P}{k}}.$$

Equation 1: The probability that two nodes share at least q keys with each other is therefore

$$1 - \sum_{i=0}^{q-1} p(i).$$

Equation 2

Protocol Specification:

The specifications for our protocols are now described in detail. For simplicity, the protocol for a network in an infrastructure less environment will be referred to as LION. The scheme relying upon the presence of the KDC will be referred to as TIGER. WIGER covers the integration of these two components.

The highlights of the protocol operation follow: All nodes are loaded with a random set of keys drawn from a common pool before being deployed. In addition, the mapping of keys to nodes is stored in a KDC. If the network is operating in stand-alone mode, i.e., with no KDC, we define a protocol to instantiate probabilistic keying. If the network has access to a KDC, we leverage the knowledge of the pre-deployed keys to perform probabilistic authentication with a high degree of confidence. In addition, session keys are established with the enforcement of least privilege. Nodes gather information in this mode of operation so that they may continue to perform some level of probabilistic authentication if the KDC becomes unavailable for periods of time. The mode of operation may change between stand-alone and KDC-mode.

The portrait of Sensor Networks painted by most of the current literature is one of extremes. Systems those exist either in total separation from infrastructure and intervention or with constant access to such resources. Networks designed to operate in isolation therefore never consider harnessing new resources as they become available. Likewise, systems designed with reliance upon available infrastructure flounder in its absence. In reality, large-scale Sensor Networks will have to optimally perform their missions in both of the above settings. If, for example, there is a method of transmitting data from a sink node to some external destination, then the ability of a Sensor node to communicate with a KDC is entirely realistic. Indeed, if data cannot be drawn out of a Sensor Network and delivered to some distant location, the usefulness of the network is extremely limited.

Simply placing only LION, TIGER, or any other single-mode scheme in a Sensor node therefore fails to fully utilize the potential of these networks. The combination of slightly modified versions of these two schemes results in WIGER—a more robust method of key management for heterogeneous Sensor Networks. The combination enables different levels of probabilistic authentication without increasing memory requirements of the L1 Sensor nodes

VII. NOTATION

- A, B are principles (e.g., communicating L1 nodes).
- $ID_{A_0}, \cdots, ID_{A_{k-1}}$ are the sorted key identifiers corresponding to the keys held by node A.
- K_A is a secret key known by node A.
- $K_{A,B}$ is a session key shared between nodes A and B.
- K_{A_AUTH} is an authenticator key for node A.
- K_{A_i} is some key corresponding to an ID from within the range described directly above.
- $L1$ is a sensor node.
- $L2(GW)$ is the L2/Gateway node.
- $MAC(K_A, R|S)$ is a Message Authentication Code of the values R and S, using key K_A.
- MAP_A is the bitmap corresponding to a sorted representation of $ID_{A_0}, \cdots, ID_{A_{k-1}}$.
- N is a nonce.
- $\{S\}_{(K)}$ is a value S encrypted in key K.

LION: Standalone Key Management

After deployment under the LION protocol, an L1 node learns its neighbors through the exchange of Hello messages2 and then attempts to establish keys with them. To accomplish this, the node broadcasts its entire key identifiers. Because the keys themselves are not transmitted and similar information could be gathered from a traffic analysis attack [9], this method does not compromise the integrity of the node itself. If a neighboring node overhears this transmission and determines that it shares one of the keys associated with the broadcast, it responds to the source with a challenge/response. Other methods of determining the keys of neighbors include the use of hash functions, encrypted broadcasts, and the use of polynomials. Messages 1 and 2 in Fig. 4 show the message flow for the case in which a node, A, has a key match, KAi , with an L2 node. The messages exchanged between the two exhibits the following format.

$$1.\ A \rightarrow * : A, N, ID_{A_0}, \cdots, ID_{A_{k-1}}.$$
$$2.\ L2 \rightarrow A : A, L2, N, ID_{A_i}, \{ID_{A_i}, L2, N\}_{(K_{A_i})}.$$

L1 nodes amass a list of neighbors with which they do and do not share keys. When the shared-key discovery phase ends, a node attempts to use the neighbors with which keys are already shared to assist it in establishing secure connections with all neighbors. In the Limited Trust model discussed here, this “Request for Assistance” (which contains the entire node IDs with which a secure relationship has not been established) is sent directly to an L2 node. The L2 node, having already established a link with the targeted L1, transmits a message to the requester and targeted node containing a session key encrypted in each of the keys shared with the L2 node. Each L1 node then receives the L2 broadcast, decrypts the session key, and begins the secure transmission of data. The messages for the indirect phase are:

$$3.\ A \rightarrow L2 : A, B, N.$$
$$4.\ L2 \rightarrow * : A, B, N, \{K_{A,B}, N\}_{(K_{A,L2})}, \{K_{A,B}, N\}_{(K_{B,L2})}.$$

Messages 3 and 4 in Fig. 4 show the indirect phase of the protocol with the “Request for Assistance” message transmitted to the neighboring L2 node following the Limited Trust model. This message would be broadcast to all neighbors if a less stringent trust model was in effect.

If a node assists in establishing a session key during the indirect phase of the protocol, it deletes this key as soon as end-to-end communication is established. The two endpoints of communication also re-key immediately. If a node is compromised at a later time, it will not contain any valid session keys other than its own.

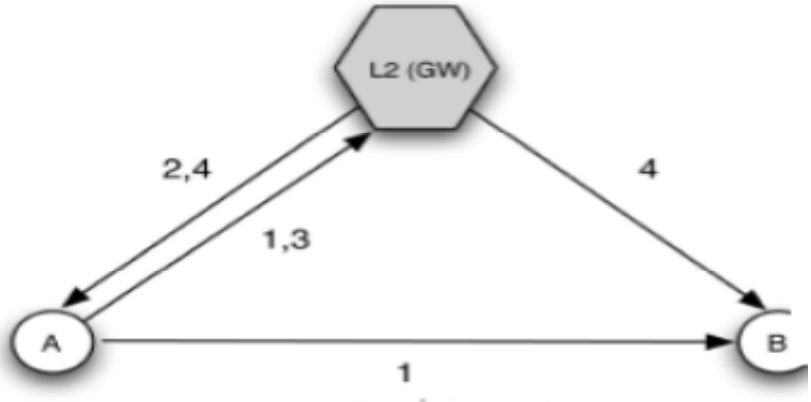


Fig. 4. LION key establishment. First, L1 node A broadcasts identifiers. A neighboring L2 determines that it has a match and sends a challenge/response message.

TIGER: KDC-Based Key Management

In locations such as "smart buildings" or factories where Sensor Networks may be used to gather data corresponding to changing environmental, structural, and inventory related conditions, access to a KDC is an entirely realistic assumption. We have designed TIGER for this scenario.

TIGER takes advantage of a KDC with a protocol enforcing least privilege over key establishments while retaining the ability to operate should the connection to infrastructure cease to exist. We discuss the case in which nodes are activated and a link to the KDC is available through an L2 node. If conditions prevent a connection being established, as is the case in the military deployment example, the network defaults to the LION protocol until a KDC link becomes available.

The messages for this protocol flow as shown in Fig. 5 and appear as follows

1. $A \rightarrow B : A, B, N, MAP_A, MAC(K_{A\_AUTH}, A|B|N|MAP_A)$.
2. $B \rightarrow L2 : A, B, N, MAP_B, MAC(K_{B\_AUTH}, A|B|N|MAP_B)$, $MAP_A, MAC(K_{A\_AUTH}, A|B|N|MAP_A)$.
3. $L2 \rightarrow KDC$: Forward Message 2 to KDC.
4. $KDC \rightarrow L2 : A, B, N, MAP_{A'}, \{K_{A,B}, N\}_{\langle K_{A'\_AUTH}\rangle}$, $MAP_{B'}, \{K_{A,B}, N\}_{\langle K_{B'\_AUTH}\rangle}$, $MAC(K_{A,B}, A|B|N|K_{A,B}|MAPA'|MAP_{B'})$.
5. $L2 \rightarrow A : A, B, N, MAP_{A'}, \{K_{A,B}, N\}_{\langle K_{A'\_AUTH}\rangle}$, $MAC(K_{A,B}, A|B|N|K_{A,B}|MAP_{A'})$.
6. $L2 \rightarrow B : A, B, N, MAP_{B'}, \{K_{A,B}, N\}_{\langle K_{B'\_AUTH}\rangle}$, $MAC(K_{A,B}, A|B|N|K_{A,B}|MAP_{B'})$.

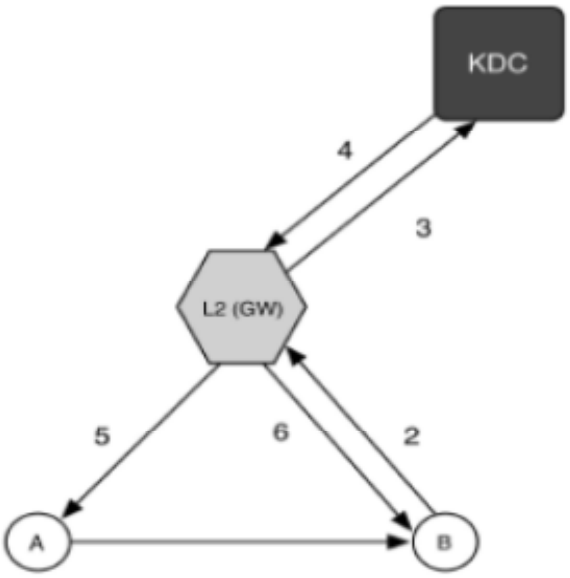


Fig. 5. The TIGER message flow for key establishment between L1 nodes. Node A sends a token to Node B. B includes its own token and forwards that message to the KDC. The KDC validates both tokens and returns a copy of the session key encoded in one of the allowable authenticator keys to both A and B.

The messages to implement protocol in other way follow the flow shown in Fig. 6 and use the format below: 6 and use the format below:

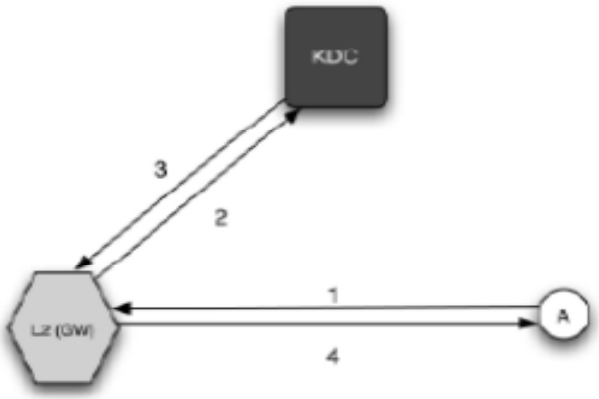


Fig. 6. The TIGER message flow for establishing keys between L1 and L2 nodes. By requiring the L2 node to include a token generated by an L1, least privilege is enforced.

$$1.\ A \rightarrow L2 : A, L2, N, MAP_A, MAC(K_{A\_AUTH}, L2|N|MAP_A).$$
$$2.\ L2 \rightarrow KDC :\ \text{Forward Message 1 to KDC.}$$
$$3.\ KDC \rightarrow L2 : A, N,$$
$$MAP_{A'}\{K_{A,L2}, N, \{K_{A,L2}, N\}_{(K_{A\_AUTH})}\}_{(K_{KDC,L2})},$$
$$4.\ L2 \rightarrow A : A, N, MAP_{A'}\{K_{A,L2}, N\}_{(K_{A'\_AUTH})},$$
$$MAC(K_{A,L2}, A|N|MAP_{A'}, \{K_{A,L2}, N\}_{(K_{A'\_AUTH})}).$$

Switching Modes of Operation

The advantage of WIGER is that it allows a Sensor Network to operate in a secure and efficient manner regardless of the available resources. There are, however, a number of tradeoffs experienced by a system operating in either mode. A comparison of these issues is made below so as to further clarify the effectiveness of the mechanisms provided by both LION and TIGER.

Specifically, we examine the effects of transitioning between modes and discuss how security is affected.

TIGER to LION: An example system likely to initialize using TIGER and transition into the LION protocol is a Sensor Network that is deployed in support of a planned operation. In this case, session keys may be initialized in a controlled environment with access to a KDC. As the operation progresses, it is possible that access to the KDC is lost. In the ideal setting, a Sensor Network is allowed to initialize in the presence of KDC. Every node in the network is able to authenticate each of its neighbours to the full extent supported by this system. Because the KDC knows all of the keys stored in both the L1 and L2 nodes, it can send the key identifiers common to an L1/L2 pair to an L2 node in message 4 of the TIGER protocol flow. If the system later transitions in to the LION protocol, either by design or out of necessity, the stored, authenticated key identifiers now in the L2 node can be used for performing authentication of L1 nodes when refreshing expired keys or helping to establish an authenticated connection between two L1s without the presence of a KDC.

VIII. CONCLUSION

In this paper, we have leveraged the emerging trend of heterogeneity in Sensor Networks to provide new, more efficient mechanisms for secure communications. We began by introducing the probabilistic unbalanced key distribution. In this scheme,

nodes with more intrinsic resources are responsible for a greater proportion of the communications and memory overhead associated with security. Whether in the presence or absence of a backhaul link, nodes running the WIGER suite are able to take advantage of all available security resources in potentially dynamic environments.

This paper also analyzes the intrusion detection problem in both homogeneous and heterogeneous WSNs by characterizing intrusion detection probability with respect to the intrusion distance and the network parameters (i.e., node density, sensing range, and transmission range).

Two detection models are considered: single-sensing detection and multiple-sensing detection models. The analytical model for intrusion detection allows us to analytically formulate intrusion detection probability within a certain intrusion distance under various application scenarios. Moreover, we consider the network connectivity and the broadcast reach ability in a heterogeneous WSN. Our simulation results verify the correctness of the proposed analytical model. This work provides insights in designing homogeneous and heterogeneous WSNs and helps in selecting critical network parameters so as to meet the application requirements.

Balancing object detection quality and network lifetime is a challenging task in Sensor Networks. Under partial coverage, we develop an analytical model for object detection applications, and mathematically study average-case object detection quality of the random and synchronized sensing scheduling protocols with respect to various network conditions. Aiming to achieve bounded worst-case object detection quality, we propose and analyze three wave sensing scheduling protocols, and formally prove the bounds on worst-case object detection quality of these protocols. Our proposed protocols and their analyses characterize the interactions among network parameters, average-case and worst-case object detection quality, and energy consumption of the protocols.

References

1. T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, “An Energy-Efficient Surveillance System Using Sensor Adhoc Networks,” Proc. ACM/USENIX MobiSys ’04, pp. 270-283, June 2004.
2. A. Mainwaring, R. Szewczyk, D. Culler, and J. Anderson, “Sensor Adhoc Networks for Habitat Monitoring,” Proc. ACM Workshop Sensor Adhoc Networks and Applications ’02, pp. 88-97, Sept. 2002.
3. A. Liu and P. Ning, “TinyECC: Elliptic Curve Cryptography for Sensor Networks (Version 0.1),” http://discovery.csc.ncsu.edu/ software/TinyECC/, 2005.
4. D. Malan, M. Welsh, and M. Smith, “A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography,” Proc. First IEEE Int’l Conf. Sensor and Ad Hoc Comm. And Networks, 2004.
5. R. Needham and M. Schroeder, “Using Encryption for Authentication in Large Networks of Computers,” Comm. ACM, vol. 21, pp. 993-999, 1978.
6. J. Kohl and B. Neuman, The Kerberos Network Authentication Service (V5), IETF RFC 1510, 1993.
7. A. Fox and S. Gribble, “Security on the Move: Indirect Authentication Using Kerberos,” Proc. MobiCom, 1996
8. L. Eschenauer and V. Gligor, “A Key Management Scheme for Distributed Sensor Networks,” Proc. ACM Conf. Computer and Comm. Security (CCS ’02), Nov. 2002.
9. B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, “Mobility Improves Coverage of Sensor Networks,” Proc. Sixth ACM Int’l Symp. Sensor Networking and Computing (WSN ’05), pp. 300-308, 2005.
10. Z. Abrams, A. Goel, and S. Plotkin, “Set k-Cover Algorithms for Vol 1 Issue 2, December 2012 141 Energy Efficient Monitoring in Sensor Networks,” Proc. Information Processing in Sensor Networks Conf. (IPSN ’04), pp. 424- 432, Apr. 2004.
11. P.B. Godfrey and D. Ratajczak, “Naps: Scalable, Robust Topology Management in Wireless Ad Hoc Networks,” Proc. Information Processing in Sensor Networks Conf. (IPSN ’04), pp. 443-451, Apr. 2004.
12. C. Hsin and M. Liu, “Network Coverage Using Low Duty-Cycled Sensors: Random & Coordinated Sleep Algorithms,” Proc. Information Processing in Sensor Networks Conf. (IPSN ’04), pp. 433-442, Apr. 2004
13. B. Liu and D. Towsley, “A Study on the Coverage of Large-Scale Sensor Networks,” Proc. IEEE Int’l Conf. Sensor Ad-Hoc and Sensor Systems (MASS ’04), pp. 475-483, Oct. 2004.
14. T. Yan, T. He, and J. Stankovic, “Differentiated Surveillance for Sensor Networks,” Proc. ACM Conf. Embedded Networked Sensor Systems (SenSys ’03), pp. 51-62, Nov. 2003

AUTHOR(S) PROFILE

Bandaru srinivasa rao studied B.Tech from jntuk, in computer science & engineering in 2012.

Yuvaraju Chinnam working as Assoc. Professor in Sri Sai Madhavi Institute of Science & Technology Rajanagaram, And doing his P.Hd in jntuk.