Gazi University Journal of Science
GU J Sci
25(2):465-476 (2012)

ORIGINAL ARTICLE

Performance Evaluation of Key Management Schemes in Wireless Sensor Networks

Önder KHALIL[1,♦], Suat ÖZDEMİR[1]

[1] *Gazi University, Computer Engineering Department, Maltepe Ankara*

Received: 24.07.2011 Revised: 30.11.2011 Accepted:04.02.2012

Abstract

Wireless sensor networks are being deployed in wide variety of applications, including military sensing and tracking, environment monitoring, patient monitoring and tracking, smart environments, etc. When a wireless sensor network is deployed in such hostile environment, security becomes an extremely important issue. Confidentiality, integrity, and availability are typical security goals for wireless sensor networks. Providing these goals to secure communication among sensor nodes typically depends on the use of cryptographic schemes. When employing a cryptographic scheme, a key management service is always required. The objective of this paper is to evaluate the most important key management schemes in wireless sensor networks which are single network-wide key scheme, pairwise key establishment scheme, random key predistribution, and Q-composite random key predistribution scheme. The evaluation is performed in OMNET++ simulation environment and the metrics are selected as secure connectivity achievement, memory overhead, communication overhead, and resilience against node capture attacks. Based on the simulation results, the advantages and disadvantages of each scheme are presented. The simulation results show that there is no general purpose key management scheme that can fit all the security requirements of wireless sensor networks. However, in terms of the performance metrics, the most suitable scheme for wireless sensor networks is the random key predistribution scheme.

Key words: key management, security, performance evaluation, analysis, wireless sensor networks.

1. INTRODUCTION

A recent technology review indicates that sensor technology is one of the ten emerging technologies that will change the world [1]. Developments in sensor network technology accelerated the deployment of Wireless Sensor Networks (WSNs) which usually consist of a large number of ultra-small autonomous devices. Each device, called sensor node, is battery powered and equipped with integrated sensors, a data processing unit, and a short-range radio communication unit. Sensor nodes are significantly constrained in terms of energy, memory, and computational capacity [2]. Figure 1, adopted from [5] depicts a schematic diagram of a sensor nodeồs components. Basically, each sensor node is composed of a sensing, processing, transmission and power units (some of these components are optional, such as the mobilizer) [3].

♦Corresponding author, e-mail: onderameen@gazi.edu.tr

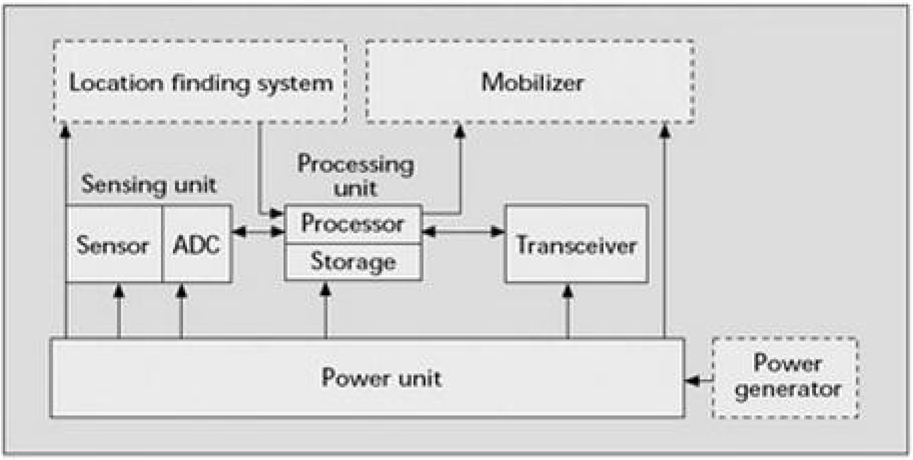


Figure 1. The components of a typical wireless sensor node [5].

In WSNs, sensor nodes are generally deployed randomly to the field of interest. The deployment environment may be on land, underground, or underwater [4]. Using wireless communication, sensor nodes form a network to collaborate on sensing the physical environment at unprecedented resolution, improving sensing quality and enabling new applications. The sensor nodes collected the data, perform data aggregation and then send the result to the sink (or base station) as can be seen in Figure 2**.**

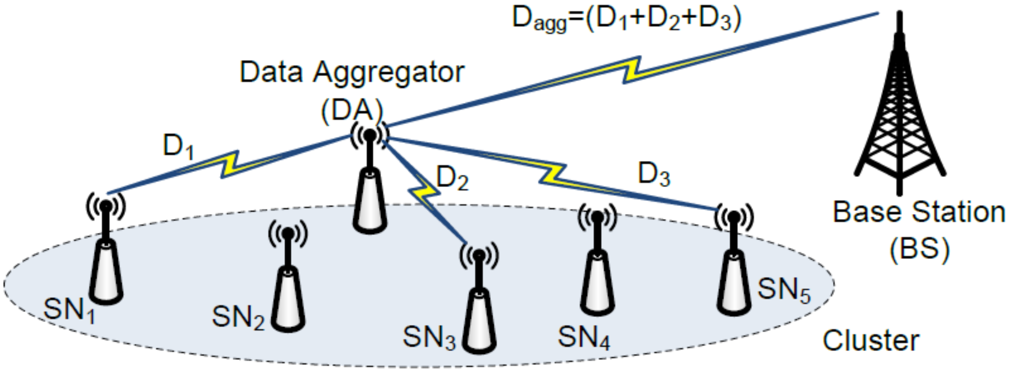


Figure 2. Sensor nodes scattered in a sensor field.

Sensor nodes in WSNs can be used to gather and process data from the environment (e.g., mechanical, thermal, biological, chemical, and optical readings), enabling many applications such as environment and habitat monitoring, support for logistics, health care and emergency response, as well as military operations [6,7]. These networks usually deployed and left in an unattended area for a long time.

Due to their unattended nature, WSNs pose security and privacy challenges. In some applications, sensor nodes have to be deployed in hostile environments and hence are subject to various external and internal attacks. For example, an adversary can easily gain access to mission critical information by monitoring wireless communications among sensor nodes, or inject false messages into the networks through some compromised nodes. Therefore, it is crucial to deploy secret keys into WSNs to encrypt wireless communications or establish authentication among sensor nodes. The challenge is how to efficiently generate, distribute and maintain secret keys among sensor nodes. This problem is called key management problem for WSNs and can be solved by carefully designed key management schemes. Traditional key distribution schemes cannot be directly used in WSNs due to their unique properties [8].

When designing a key management scheme for WSNs, designers should take the following five major resource constraints of sensor nodes into consideration: (1) limited energy, (2) limited memory, (3) limited computing power, (4) limited communication bandwidth, (5) limited communication range [8]. In addition to these constraints, there is also lack of physical security of sensor nodes. WSNs are deployed in unattended and hostile regions, and therefore physical security of sensor nodes cannot be guaranteed. The lack of physical security results in node capture attacks where an attacker gains the control of a node in the network after deployment. Once in control of that node, the attacker can maliciously alter the node to listen to information in the network, input false data, and perform various attacks on the network. The attacker may also simply obtain the information critical to the networkôs security such as routing protocols, data, and security keys [10]. Hence, key distribution schemes of WSNs must consider the compromised nodes as well.

The objective of key management is to establish and maintain secure and dynamic channels among communicating nodes [9]. The desired features of key management scheme can be summarized as follows:

Scalability: Efficiency demands that WSNs utilize a scalable key management scheme to allow for variations in the size of the network. Key management schemes should provide their features for small size networks, but also maintain these characteristics when applied to larger ones.

Flexibility: Key establishment techniques should be able to function well in any kind of environments and support dynamic deployment of nodes, i.e., a key establishment technique should be useful in multiple applications and allow for adding nodes at any time.

Memory: Memory availability of sensor nodes is usually 6ï 8 Kbps, half of which is occupied by a typical sensor network operating system. Key establishment techniques must use the remaining limited storage space efficiently by storing keys in memory, buffering stored messages, etc.

Key management schemes of WSNs should take into consideration all the aforementioned requirements and constraints. This paper investigates the most important key management schemes in WSNs. Specifically, single network-wide key scheme, pairwise key establishment scheme, random key predistribution scheme, and Q-composite random key predistribution scheme are explained in detail. The paper also presents the results of the extensive performance evaluation of these schemes in terms of communication and memory overhead, resilience to node capture and secure connectivity.

The rest of the paper organized as follows. In Section 2, the single network-wide key, pairwise key establishment, random key predistribution, and Q-composite random key predistribution schemes are explained in detail. Section 3 presents the performance evaluation and comparison of these schemes. Related wok is presented in Section 4. Finally, concluding remarks are made in Section 5.

2. Background

This section explains the key management schemes that are evaluated in this paper. Single network-wide key and pair-wise are straightforward and easy to understand scheme, hence we explained them briefly. Random key predistribution schemes on the other hand are explained in detail.

2.1 Single network-wide key

Single network-wide key management is the simplest key management technique. In the initialization phase of this technique, a single key is preloaded into all the nodes of the network (Figure 3). After deployment, every node in the network can use this key to encrypt and decrypt messages [8,9]. Some of the advantages offered by this technique include minimal storage requirements and avoidance of complex protocols [9]. Only a single key is to be stored in the nodesô memory and once deployed in the network, there is no need for a node to perform key discovery or key exchange since all the nodes in communication range can transfer messages using the key which they already share [22]. However, this scheme has a major security loophole. If one of the sensor nodes is compromised, then the adversary obtains the network wide key and communication security of the network collapses. Hence in this scheme, neighboring sensor nodes must establish pairwise keys right after the network deployment.

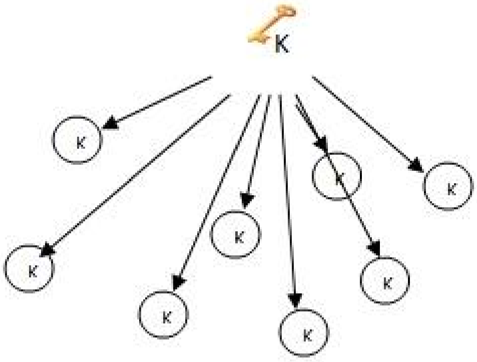


Figure 3. Single network-wide key scheme

2.2. Pairwise key establishment scheme

This scheme offers many additional features including node-to-node authentication and resilience to node replication. Hence, in a network of n nodes, there are a total (2^n) of unique keys. As shown in Figure 4., every node stores $(n-1)$ keys, one for each of the other nodes in the network. i.e., pairwise keys, which are retained in each nodeôs memory so that each node can communicate with all the nodes in its communication range [15].

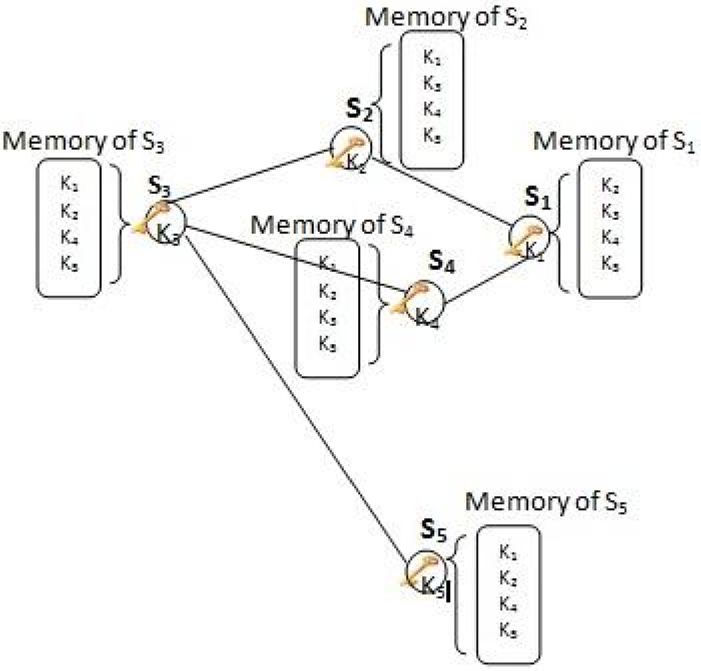
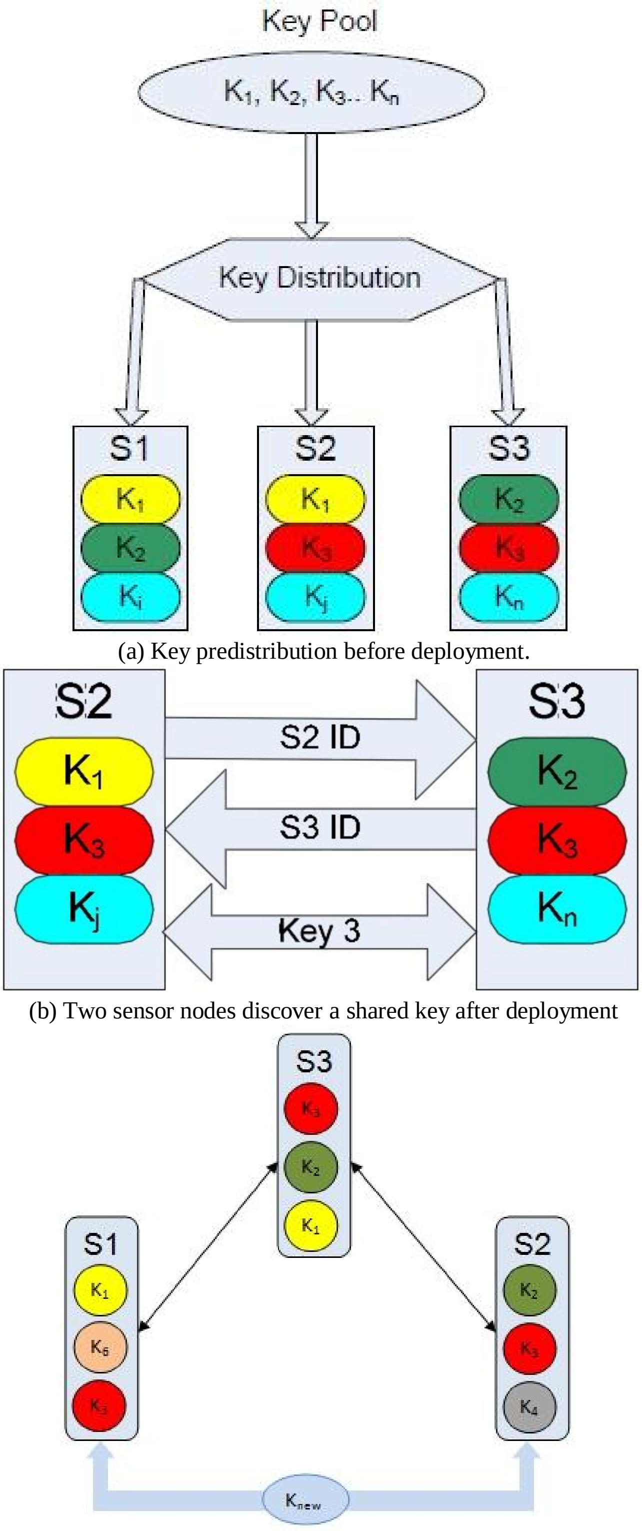


Figure 4. Pairwise key establishment scheme

This scheme provides increased resilience to network capture as a compromised node does not reveal information about other nodes that are not directly communicating with the captured node. In addition, through increased resilience, the scheme minimizes the chance for node replication [22]. However, memory overhead of this scheme is significantly higher than single network wide key approach. Since each sensor node has a distinct pairwise key for every other node in the network, the pairwise key approach is not scalable for large WSNs.

2.3. Random key predistribution scheme (Basic scheme)

In random key predistribution schemes, first a small amount of keys are preloaded into each sensor, and then these sensor nodes are deployed into the area to be monitored. Since the preloaded keys are randomly selected from a key pool, after the deployment, a sensor node can discover other nodes that it shares a key. Sensor nodes that do not have any shared key can establish a secret key using other nodes that they share a key. This scheme improves pairwise key establishment scheme by reducing the number of keys stored in each sensor node. As shown in Figure 5 (a), (b), and (c), key predistribution schemes consist of three phases [21], namely, key predistribution phase, shared key discovery phase and secure path key establishment phase. In key pre distribution phase, only a few keys need to be stored in each nodeȏs memory and these few keys are enough to ensure that two nodes share a common key based on a predetermined probability. In shared key discovery phase, each node discovers its neighbors in its wireless communication range that it shares a common key. Path key establishment phase is facilitated to assign a secure communication link between two neighboring nodes that do not share a common key. In this phase, two neighboring sensor nodes that do not share a key discover a secure path between them. A secure path is a path on which each consecutive node pair shares a common key [16].



(a) Key predistribution before deployment.

(b) Two sensor nodes discover a shared key after deployment

(c) Secure path key establishment using another node

Figure 5. Random key predistribution scheme

Random key predistribution scheme is initially proposed by Eschenauer and Gligor [16] and later on several variations of this scheme are proposed [17]. In this scheme, each sensor node store a random subset of k key from a large key pool $|S|$ before the deployment. The number of keys in the key pool,, is chosen such that two random subsets in S can share at least one key with some probability p [4]. The scheme is based on the random-graph theory [23]. A random graph $G(n,p)$ is defined as a graph of n vertices, in which the probability that a link exists between two nodes is p . In the network graph two nodes are adjacent if they share a secret key. The graph connectivity P_c has the following relation with p [16]:

$$P_c = \lim_{n\to\infty} P_c[G(n,p) \; is \; connected] = e^{e^{-c}}$$

where $p = \frac{ln(n)}{n} + \frac{c}{n}$ and c is a constant.

It follows that given n we can find p and the expected degree of a node $p = p*(n-1)$ in which the resulting graph is connected with required probability P_c. Also the wireless connectivity constraints may limit the number of neighboring nodes to $n' \ll n$ nodes, then the probability of sharing a key between any two nodes in a neighborhood becomes

$$p' = \frac{d}{n'-1} \gg p$$

This scheme works as below.

1. Choose c for a desired probability of connectivity P_c such that $P_c = e^{e^{-c}}$
2. Calculate p by $p = \frac{\ln(n)}{n} + \frac{c}{b}$ and
3. Determine the size of key pool S and the size of key ring k . Such that S and k should satisfy

Thus, a key pool of size *S* is defined and each node can randomly select *k* keys in the key pool.

2.4. Q-composite random key predistribution scheme

This scheme is an extension of random key predistribution scheme and it enhances the security and resilience of the network against node capture attacks. In Q-composite random key predistribution scheme, in order to establish a secure communication link, a sensor node pair must share at least q keys where q is a system parameter and q > 1 [22]. Q-composite scheme achieves security under small scale attacks while being vulnerable under large scale attacks. The major challenge of this scheme is to select an optimal value for *q* while ensuring that security is not sacrificed. If the amount of key overlap between two nodes is large (i.e., large value of q), it becomes harder for an adversary to break the communication link, at the same time this means that by compromising a small amount of sensor nodes the adversary can gain a large part of key pool that is used by sensor nodes.

Similar to [16], in the key pre distribution phase, each sensor node picks k random keys from $|S|$ and initialized in each nodeôs key ring. In the shared key discovery phase each node must discover all common keys it possess with each one of its neighbors. This can be accomplished with a simple local broadcast of all key identifiers that the node possess. A more secure but a slower method of shared key discovery is using client puzzles such as a Merkle Puzzle [17]. S , the size of the key pool, is the critical parameter that must be calculated for the Q-composite scheme to be efficient and secure. If S is large, then the probability that two nodes share a common key and therefore can communicate is decreased. However, if S is decreased, an adversaryôs job may be easier as she can gather most of the keys in the key pool by capturing only a few nodes. Thus, S must be chosen such that the probability of any two nodes sharing at least q keys is is larger than or equal to p . As defined in [17], S can be calculated as follows

$$p(i) = \frac{\binom{|S|}{i}\binom{|S|-i}{2(m-i)}\binom{2(m-i)}{m-i}}{(m^{|S|})^2}$$

where $p(i)$ is the probability that any two nodes have exactly i number of keys in common; and m is the key ring capacity for a given node. There are $\binom{|S|}{i}$ ways to pick i and $|S| - i$ is the number of the remaining keys in the key pool after *i* is picked. There are $\binom{|S|}{m}$ different ways to pick m and $\binom{|S|}{m}^2$ total number of ways for both nodes to pick m . Also, to assign the remaining keys $2(m-i)$ distinct keys are picked from the key pool for each node and the number of ways to do this is $\binom{|S|-i}{2(m-i)}$. There are $2(m-i)$ ways to partition the keys equally between the two nodes. Let P_c be the probability of any two nodes sharing sufficient keys to form a secure connection. Therefore, $P_1c = 1 - ($ the probability that the two nodes share insufficient keys to form a connection) or $P_c = 1 - \left(p(0) + p(1) + \cdots + p(q-1)\right)$. Now the largest $|S|$ such that $P_c \geq p$ is chosen.

3. Performance Evaluation and Results

In this section, to evaluate the security and performance of the key distribution schemes, we have taken the following metrics into consideration.

Secure connectivity: Secure connectivity shows the ratio of securely connected links to all links in the network. For a key management scheme, higher secure connectivity can be achieved by either having large number of node pairs that share a secret key or offering an efficient and secure path key establishment method.

Memory overhead: Since memory of sensor nodes is mainly occupied by operating system and application programs, the remaining part should be used carefully. Hence, a key management scheme should be as efficient as possible in terms of the numbers of keys that has to be stored in a sensor node.

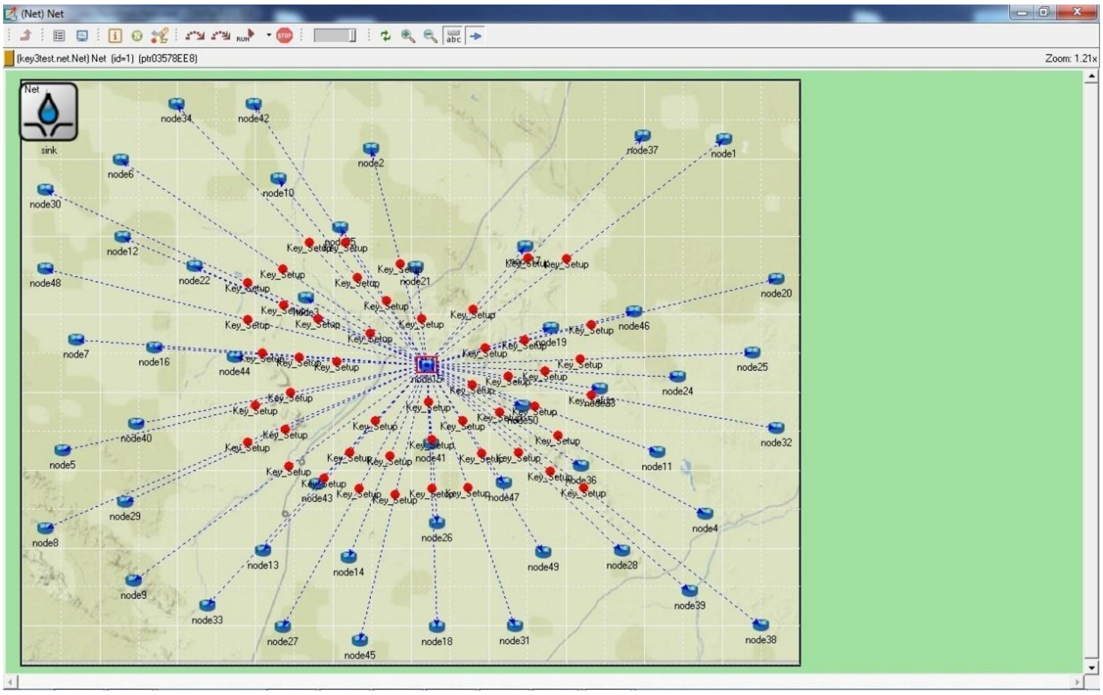
Communication overhead: As the transmission is the major source of the battery consumption, communication required by a key management scheme must be small. In addition, transmitting the secret information over the air increases the security threats. Hence, a key management scheme should not incur high communication overhead.

Resilience against node capture: As WSNs are usually deployed in unattended regions, it is always possible to mount a physical attack on sensor nodes. Once a sensor node is compromised, its secret keys are obtained by the adversary as well. Hence, key management schemes must ensure that a compromised sensor node does not reveal too much information to the adversary.

3.1 Simulation environment and scenarios

In order to evaluate the key management schemes discrete event based OMNET++ simulator [24] is employed. OMNET++ ñObjective Modular Network Test-bed in C++ò is one of the most attractive network simulators [25]. Its primary application area is the simulation of communication networks, however, because of its generic and flexible architecture, it has been successfully used in other areas such as modeling of multiprocessors and other distributed hardware systems, validating of hardware architectures, and evaluating performance aspects of complex software systems. The OMNeT++ model consists of hierarchically nested modules. The top-level model is the system model, which encompasses the complete simulation model and is referred to as the ñnetworksò. The system contains sub-modules which themselves may have sub-modules [26]. Modules that contain sub-modules are called compound models. The user implements the simple modules in C++, using the OMNeT++ simulation class library [27]. Modules communicate by message passing which may be a complex data structure. Modules may send messages directly to their destination or through a series of gates and connections to other modules [28]. As a hierarchal model is followed, the messages typically travel through a series of connections that start and end at simple modules.

As all of our performance metrics are affected by the network size, simulations were performed for different network sizes (10, 25, and 50 nodes). The area of each network was and each network was implemented independently from other two. The medium access scheme is selected as CSMA and the default bit error rate was 10%. The key pool size are chosen to be 100 key and key ring size is set to 3. The key pool size and selected key rings ensure 90% secure connectivity among sensor nodes for both random key predistribution and Q-composite schemes. In Figure 6 a screenshot of an example simulation scenario is given.


(Net) Net

[key3test.net.Net] Net (id=1) (ptr03578EE8)

Zoom: 1.21x

Figure 6. Screenshot of OMNET++ simulation environment

3.2 Simulation Results

In this section, we present the simulation results in terms of secure connectivity, memory overhead, communication overhead, and resiliency to node capture.

3.2.1 Secure connectivity:

We measured the secure connectivity rate of the network size of 50 nodes (as the size of the network has no impact of the overall performance the result given only for the case of 50 nodes) without path key establishment phase and present the results in Figure 7.

As seen from Figure 7, in single network- wide key scheme since there is single key in the network each node can communicate with any node that falls into its communication range making the secure connectivity rate 100% . In the pairwise key establishment scheme the connectivity is also 100% , since each node carries $n-1$ keys for every other node in the network. In random key predistribution scheme each node connects to other nodes with the probability of p (where $p = 0.67$). Simulation results also show that the secure connectivity rate of the network is 70% which is determined by the value of p . Similarly, in Q-composite random key predistribution scheme each sensor node can connect to one of its neighbors with a $p(q)$ probability (where $p(q) = 0.9$). In the simulation, q is selected as 3 and the network shows 23.33% secure connectivity rate.

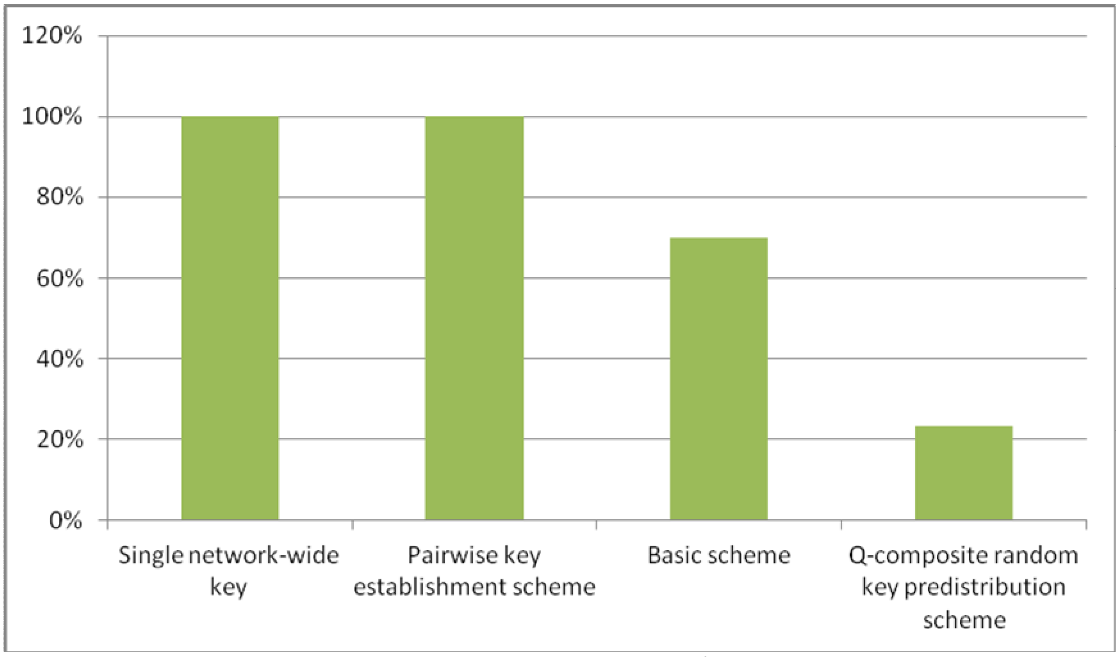


Figure 7. Secure connectivity rate of the network

3.2.2 Memory overhead

Figure 8 presents the memory overhead of each key management scheme for different network sizes. The results shows that single network- wide key scheme uses only 4 bytes of memory which it is the size of one key. In pairwise key establishment scheme, since each sensor node is loaded with a distinct key for every other node in the network, this schemeȏs memory overhead per node is $(n-1) \times 4$ bytes. For example, a sensor node in a WSN consisting of 40-node incurs 36-byte memory overhead. In our simulation, basic and Q-composite schemes employ 3 keys per sensor node resulting in 12-byte memory overhead. As seen from Figure 8, this overhead is not affected by the network size.

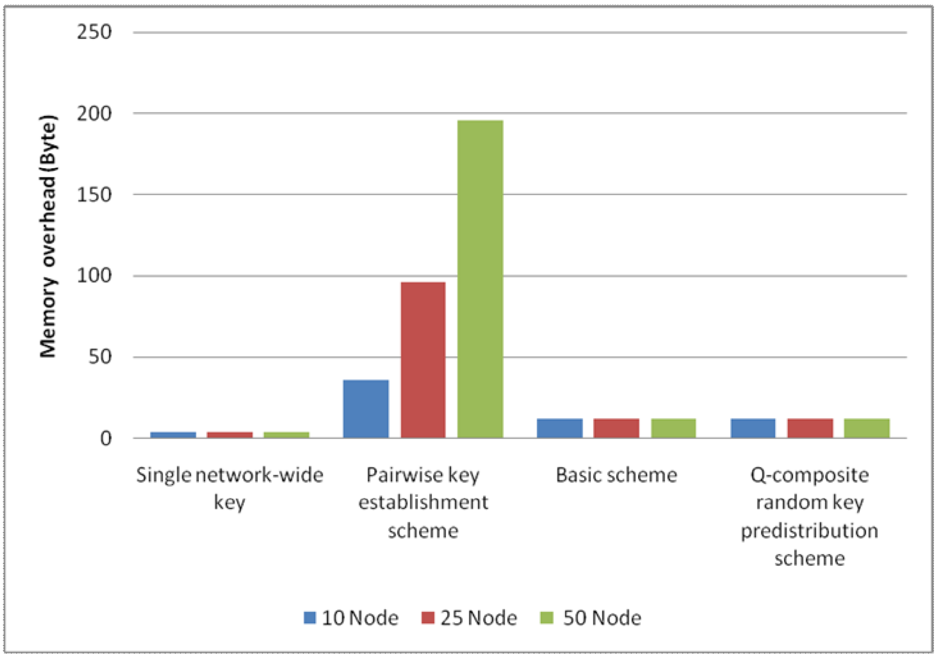


Figure 8. Memory overhead

3.2.3 Communication overhead

Figure 9 illustrates the communication overhead of each key management scheme. In single network-wide key scheme there is no communication overhead because we assign a single key to all nodes in the network before network deployment. As seen from Figure 9, random key predistribution schemes have more communication overhead than pairwise key establishment scheme as in pairwise key establishment scheme each node has single key while in the random key predistribution schemes each node has number of keys equal to the ring size of the node.

For example in the case of 50 nodes in basic random keypredistribution scheme communication overhead due to key establishment is 1800 bytes whereas in pairwise key establishment scheme this overhead is equal to only 270 bytes. Basic scheme and Q-composite random key predistribution scheme have almost same overhead; this is due to the fact that they both employ the same key discovery phase using small key identifiers.

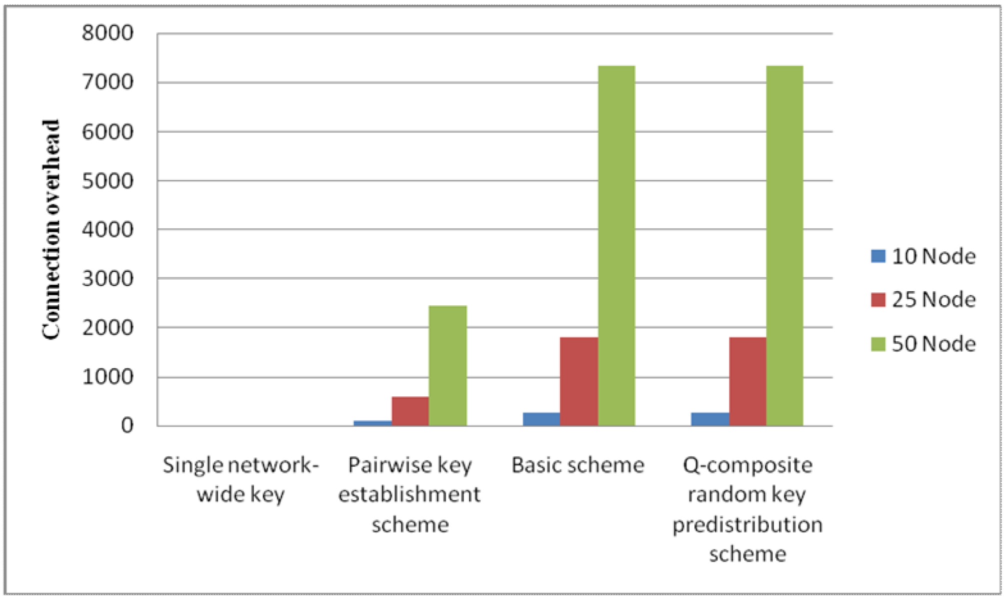


Figure 9. Communication overhead due to the key management scheme (bytes)

3.2.4 Resilience against node capture

Figure 10 presents the resiliency against node capture. The resilience is measured in terms of the number of secret keys a compromised node reveals. If a sensor nodeôs secret keys are revealed we assume that sensor node is also captured. From Figure 10, we can see that pairwise establishment scheme is the most efficient and resistant scheme while in single network-wide key it is enough to capture one node to gain control to the entire network. The simulation results also show that Q-composite scheme has better resilience compared to basic scheme.

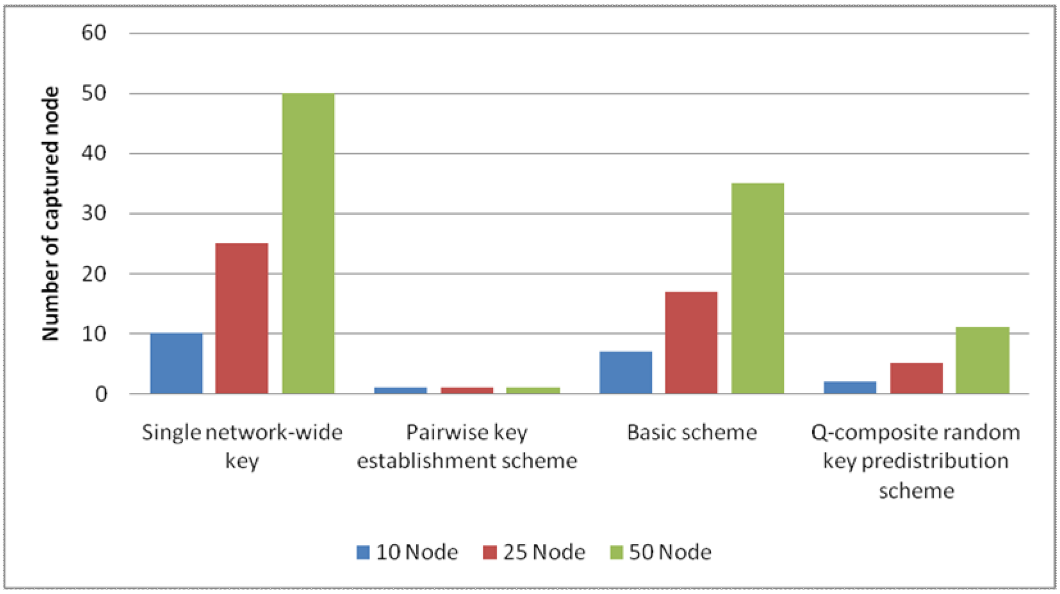


Figure 10. Resilience against node capture

4. RELATED WORK

Several key symmetric key establishment protocols have been proposed for WSNs [15] based on centralized key server or pre-distributed secret keys. Key Distribution Center (KDC) based schemes depend on the existence of a key distribution center which is trusted and has no resource limitation. KDC acts as the arbiter for key establishment process between two sensor nodes [15]. Memory overhead of KDC based protocols is very low as each node only needs to secure its communications with the KDC. However, communication overhead is high since each node has to communicate with KDC for each key establishment.

Key establishment protocols based on pairwise keys require distribution of pairwise keys to sensor nodes before the network deployment. Full pairwise key distribution protocol where each node in a network of *n* nodes shares a unique pairwise key with every other node in the network. The communication overhead of this protocol is minimal but the memory overhead is (*n*-1)-key per node so it is not scalable. Therefore, in pairwise key distribution protocols, the locations of sensor nodes must be known so that sensor nodes are given only the pairwise keys that they need. However, in wireless sensor networks, sensor node locations are usually not known in advance.

The memory overhead of pairwise key distribution protocols is reduced in random key predistribution schemes that store secret keys in sensor nodes just enough to ensure that any two sensor nodes can perform key establishment. Such a random key predistribution scheme is proposed by Eschenauer and Gligor in [16]. In the proposed scheme, a random pool of keys is selected from the key space. Each sensor node receives a random subset of keys from the key pool before network deployment. Any two nodes able to find one common key within their respective subsets can use that key as their shared secret to initiate communication. Eschenauer and Gligor's scheme is improved in [17-19]. In [18], authors use the estimated location information of sensor nodes to reduce memory space and computational overhead due to key distribution. The key distribution scheme in [17] is very similar to Eschenauer and Gligor's scheme except that their approach requires any pair of sensor nodes to have *q* common key within their key set. The work in [19] presents a key distribution scheme based on polynomial-based key pre-distribution which reduces the computational needs of sensor nodes. Although, random key predistribution schemes provide a balanced communication and memory overhead, due to their probabilistic nature, they are only suitable for networks where the random graph model for connectivity holds.
For example, in a network where nodes are not densely distributed, or in a network where node density is non-uniform, performing probabilistic key establishment could result in a disconnected graph due to few critical sensor node pairs that could not successfully perform key establishment. Also, if a small fraction of sensor nodes are compromised by the same intruder, then the amount of compromised keys could be significantly high which reduces resilience against node compromise attacks.

5. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Wireless sensor networks are being deployed in wide variety of applications, including military tracking and security is a vital requirement for these networks. Security protocols need key management schemes to establish secret keys between communicating parties. This paper investigates and evaluates the most important key management schemes in wireless sensor networks. Namely, single network-wide key scheme, pairwise key establishment scheme, random key predistribution, and Q-composite random key predistribution schemes are explained and evaluated using OMNET++ simulator. Extensive simulation results and comparisons are presented. The results show that random key predistribution schemes are the most suitable key management protocols for wireless sensor networks in terms of performance and security. Our future research directions involve comparing more key management schemes using different metrics and larger network sizes.

REFERENCES

[1] National Intelligence Council, Mapping the Global Future: Report of the National Intelligence Council's 2020 Project, (2004).

[2] Ammari H. M., ñChallenges and Opportunities of Connected k-Covered Wireless Sensor Networks From Sensor Deployment to Data Gatheringò, ***Studies in Computational Intelligence***, vol. 215, Springer, (2009).

[3] Karl H., Willig A., ñProtocols and Architectures for Wireless Sensor Networksò, ***John Wiley & Sons***, (2007).

[4] Yick J., Mukherjee B., Ghosal D., "Wireless sensor network survey," ***Computer Networks***, 52, pp. 2292ï 2330, (2008).

[5] Akyildiz I. F, Vuran M. C., ñWireless Sensor Networksò, ***John Wiley & Sons Ltd***., (2010).

[6] Alemdar, M. Ibnkahla, ñWireless sensor networks: applications and challengesò, in Proceedings of the Ninth International Symposium on Signal Processing and Its Applications (ISSPA 2007), ***IEEE Computer Society***, Washington, DC, USA, (2007).

[7] Arampatzis T., Lygeros J., Manesis S., ñA survey of applications of wireless sensors and wireless sensor networksò, in Proceedings of the 2005 IEEE International Symposium on Intelligent Control Mediterranean Conference on Control and Automation, ***IEEE Computer Society,*** Washington, DC, USA, (2005).

[8] Xiao Y., Rayi V., Sun B., Du X., Hu F., Galloway M., ñA survey of key management schemes in wireless sensor networksò, ***Computer Communications***, 30(11-12): 2314-2341, (2007).

[9] Zhang J., Varadharajan V., ñWireless sensor network key management survey and taxonomyò, ***Journal of Network and Computer Applications***, 33(2): 63-75, (2010).

[10] Hartung C., Balasalle J., Han R., ñNode Compromise in Sensor Networks: The Need for Secure Systemsò, ***Department of Computer Science University of Colorado at Boulder*** Technical Report CU-CS-990-05, Jan. (2005).

[11] Stallings W. ñCryptography and Network Security: Principles and practicesò 4th edition***, Prentice Hall***, (2006).

[12] Yong W., Garhan A., Byrav R. ñA Survey of Security Issues in Wireless Sensor Networksò ***IEEE Communications Society,*** V.8, No. 2, (2006).

[13] Su Z., Lin C., Feng F. J., Ren F. Y., ñKey Management Schemes and Protocols for Wireless Sensor Networksò, ***Journal of Software, Beijing: Science Press***, pp. 1218-1231, (2007).

[14] Silva R. M. S., Pereira N. S. A., ñChaos Based Key Management Architecture for Wireless Sensor Networksò, ***Australian Telecommunication Networks and Application Conference [ATNAC 2006]***, Melbourne, Australia, Dec. 4-6, (2006).

[15] Çamtepe S., Yener B., ñKey distribution mechanisms for wireless sensor networks: a surveyò, ***Technical report, Rensselaer Polytechnic Institute***, (2005).

[16] Eschenauer L., Gligor V., ñA key-management scheme for distributed sensor networksò, ***9th ACM Conference on Computer and Communication Security***, (2002).

[17] Chan H., Perrig A., Song D., ñRandom Key Predistribution for Sensor Networksò, ***IEEE Symposium on Security and Privacy***, (2003).

[18] Du W., Deng J., Yunghsiang S., Varshney P., ñA Pairwise Key Predistribution Scheme for Wireless Sensor Networkò, In ***Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)***, Washington D.C., Oct. 27-31, (2003).

[19] Liu D., Ning P., ñLocation-Based Pairwise Key Establishments for Relatively Static Sensor Networksò, ***2003 ACM Workshop Security of Ad Hoc and Sensor Networks (SASN03)***, Oct. 31, (2003).

[20] Du W., Deng J., Han Y., Chen S, Varshney P., ñA Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge***ò, IEEE INFOCOM 04***, Mar. 7-11, (2004).

[21] Liu D., Ning P., ñEstablishing Pairwise Keys in Distributed Sensor Networksò, To appear in the ***10th ACM Conference on Computer and Communications Security (CCS03)***, Washington D.C., October, (2003).

[22] Dong S., Bing M., Review of Key Management Mechanisms in Wireless Sensor Networks, Vol. 32, No. 6, Nov. (2006).

[23] Spencer J., ñThe Strange Logic of Random Graphs, Algorithms and Combinatoricsò, ***Springer,*** (2000).

[24] Ķnternet: OMNET++, http://www.omnetpp.org.

[25] Drytkiewicz W., Sroka S., Handziski V., Köpke A., Karl H., ñA Mobility Framework for OMNeT++ò, Jan. 22, (2003).

[26] Vargo A., Hornig R. ñAn Overview Of The Omnet++ Simulation Environmentò, Mar. (2008).

[27] Xian X., Shi W., Huang H., ñComparison of OMNET++ and other simulator for WSN simulationò, Jun. (2008).

[28] Wang S., Liu K.Z., Hu F.P., ñSimulation of Wireless Sensor Networks Localization with OMNeTò, ***Mobile Technology, Applications and Systems, 2005 2nd International Conference on***, Nov. (2005).