Key Management in Wireless Sensor Networks - A Review

Majida C V[1], Sandhya V[1]

[1]Department of Computer Science, KMCT College of Engineering, Calicut, Kerala, India

Abstract: ***Wireless sensor networks (WSN) are always prone to security threats and they do have a wide range of applications. Key management is a serious issue while communications through wireless sensor networks take place. Traditional network security methods are not up to the mark due to limited resources. Several key managements have been proposed and few of which have been summarized below. The basic method used in homogeneous sensor networks used symmetric keys which then progressed to the use of asymmetric keys. Due to the energy constraints, the concept of heterogeneous wireless sensor networks with high and low energy sensors was developed. A combination of symmetric and asymmetric keys were tried and obtained better results like security, scalability and connectivity.***

Keywords: WSN, Key management

1. Introduction

Wireless Sensor Networks (WSNs) consists of tiny devices called sensors, which are deployed to cooperatively process and communicate the data about a targeted environment. They have wide range of applications like military and national defence, medical care, environmental monitoring, wildlife tracking, weather checking applications, traffic management and in many other areas [1]. People can access to a large number of detailed, reliable information at any time, place and environment. The sensor nodes deployed in a hostile environment can be used to detect, monitor and collect the data, and to perform decision making and evaluation [2].They can be eavesdropped, captured and compromised. Sensor nodes detect enemy intrusion in battle field, measure various environmental variables and keep the information secret for which it is important to establish a secure communication between the sensor nodes. For secure communication between two sensor nodes a secret key is present [3]. Hence, WSNs security serves to conserve the confidentiality, integrity and availability of the transmitted information. Providing security is a complicated task compared to a traditional wired network or a mobile ad- hoc network due to many reasons like node's resource constraints, the wireless communication employed, deployment methods, location of the field of interest and presence of huge number of nodes in the network. These sensor nodes have low processing power, less memory capacity and less battery life. [4] Due to these limitations, WSN's have been

divided into homogeneous WSN and heterogeneous WSN. All sensor nodes in homogeneous WSN have the same capabilities and heterogeneous WSN incorporate different types of sensor nodes with different capabilities. They contain a small number of powerful high-end sensor nodes (*H*-Sensors) and a large number of low-end sensor nodes (*L*-Sensors) [5].

Key management can be defined as can be mainly classified into two categories: static and dynamic key management schemes. All keys are pre-distributed in sensor nodes in the static key management schemes. These schemes are based on the probabilistic key redistribution that guarantees a high probability of sharing keys between nodes [6]. In the dynamic key management scheme, some keys or key seeds are predistributed in sensor nodes and the session keys are established on demand. Mostly there are some nodes acting as group heads or gateways and these intermediate nodes are usually more powerful than other member nodes in terms of energy supply, transmission range, data processing capability, storage capacity, and tamper resistance [7].

2. Few Key Management Schemes

Eschenauer et.al [8] proposes a probabilistic key predistribution technique called basic scheme or E-G method. In this method, an initialization phase is performed before the deployment of sensor nodes. Basic scheme picks a random set of keys *S* out of the total possible key space. If *m* is the number of distinct cryptographic keys that can be stored on a sensor node, random selection of m keys from the key pool S is done and stored into the node's memory. This set of *m* keys is called the node's *key ring*. The number of keys in the key pool is chosen such that two random subsets of size *m* in *S* will share at least one key with some probability *p*.

Key-setup phase is performed after deployment. The nodes first perform key-discovery to find out the neighbors sharing the same key. Such key discovery can be performed by assigning a short identifier to each key prior to deployment, and having each node broadcast its set of identifiers. Nodes which discover that they contain a shared key verify that their neighbor actually holds the same key through a challenge response protocol. This shared key then becomes the key for that link.

After key-setup is complete, a connected graph of secure links is formed. Nodes can then set up *path keys* with nodes in their vicinity that did not share keys within their key rings. If the graph is connected, a path can be found from a source node to its neighbor. The source node can then generate a path key and send it securely via the path to the target node.

The highlights of the proposed method is that this method relies on probabilistic key sharing among the nodes of a random graph and also use simple protocols for shared-key discovery and path-key establishment, and for key revocation. But this method holds some disadvantages like high key-sharing probability is required and random selection of key ring from a key pool. Even large number of keys is preloaded in each sensors in order to achieve high key-sharing probability and for key pre distribution large storage space should be provided.

Haowen Chan et. al [9] investigated the *random-pairwise keys scheme*, which assures that, even when some numbers of nodes have been compromised, the remainder of the network remains fully secure. Furthermore, this scheme enables node-to-node mutual authentication between neighbors and quorum-based node revocation without involving a base station. Node-to-node mutual authentication here refers to the property that any node can ascertain the identity of the nodes that it is communicating with.

The *q*-composite keys scheme operates similar to that of the basic scheme, differing only in the size of the keypool *S* and the fact that multiple keys are used to establish communications instead of just one. In the initialization phase, the researchers picked a set *S* of random keys out of the total key space, and *m* random keys from *S* are selected (where *m* is the number of keys each node can carry in its key ring) which are then stored into the node's key ring.

In the key-setup phase, each node must discover all common keys it possesses with each of its neighbors. This can be accomplished with a simple local broadcast of all key identifiers that a node possesses. While broadcast-based key discovery is straightforward to implement, it has the disadvantage that a casual eavesdropper can identify the key sets of all the nodes in a network and thus pick an optimal set of nodes to compromise in order to discover a large subset of the key pool *S*.

After key discovery, each node can identify every neighbor node with which it shares at least *q* keys. The keys are hashed in some canonical order, for example, based on the order they occur in the original key pool *S*. Key-setup is not performed between nodes that share fewer than *q* keys.

[8], [9] methods are not preferable for large networks that uses large number of keys since wide storage space is needed. Reza Azarderakhsh et. al [10] proposed a key management method based on ECC. Cluster heads send session key of adjacent nodes with ECC. Since the session keys of adjacent nodes are different, the security of communication between other nodes when a node is captured will not be threatened. It provides a good resilience. But each ordinary node need store public keys of all cluster heads, which increases the storage consumption.

The key management method used here makes use of both private and public key cryptography. Public key cryptography is used to establish a secure link between sensor nodes and gateways. Instead of preloading a large number of keys into the sensor nodes, each node requests a session key from the gateway to establish a secure link with its neighbors after clustering phase.

Network setup phase involves Network Bootstrapping and Clustering. In the bootstrapping stage, gateways discover the nodes that are located in their communication range and broadcast a message indicating the start of clustering which is done at different instance of time in order to avoid collisions. [11]

In the clustering phase, gateways calculate the cost of communications with each node in the range set and exchange this information between all gateways. On receiving the information from all other gateways, each gateway starts clustering the sensor nodes, based on the communication cost and the current load on its cluster. When the clustering is over, all the sensors are informed about the ID of the cluster they belong to. Since gateways share the common information during clustering, each sensor node is picked by only one gateway. [12]

An intra-cluster routing scheme is used to determine how to route packets from sensor nodes to the gateway in a multihop manner. Each sensor node sends a message, including its ID, its neighboring nodes, and its location in-formation to the gateway. Gateways construct Least-Cost-Path (LCP) routing or Minimum Spanning Tree (MST) to reach sensor nodes. As the gateways are powerful and have sufficient memory resources, one broadcast is enough to cover all the sensor nodes in each cluster. Broadcast from gateways must be authenticated with sensor nodes to prevent attacks and fake messages from adversaries. In this case, broadcast authentication can be provided with, for example, the ECDSA digital signature scheme [13], without requiring time synchronization. An alternate routing algorithm is needed to ensure reliable communication among sensor nodes in emergency conditions. Sensor nodes may find different parent nodes to reach the gateway so that they can reserve them for future routing algorithm in case of any changes.

In this work secure link was established by using both private and public key cryptography and instead of preloading a large number of keys into the sensor nodes, each node requests a session key from the gateway to establish a secure link with its neighbors after clustering phase.

Drawbacks like increased storage consumption and lack of timely update of keys, chances are there for node capture and also key revocation is not considered. These drawbacks have attracted alternate methods.

Du et al. [14] proposed routing-driven elliptic curve cryptography where a node just needs to establish communication with a small portion of its neighbors called c-neighbors and does not need to setup shared keys for each pair of neighbor sensors. According to the routing information, cluster heads encrypt session keys with ECC and then sent them to adjacent nodes which need to establish communication .This method significantly reduce the overhead of key establishment, communication and computation overheads, and hence reduce sensor energy consumption.

Public-key cryptography has been considered too expensive for small sensor nodes, because traditional public-key algorithms (such as RSA) require extensive computations and are not suitable for tiny sensors. However, the recent progress on Elliptic Curve Cryptography (ECC) provides new opportunities to utilize public-key cryptography in sensor networks. The recent implementation of shows that an ECC point multiplication takes less than one second, which demonstrates that the ECC public-key cryptography is feasible for sensor networks. Compared with symmetric key cryptography, public-key cryptography provides a more flexible and simple interface, requiring no key pre-distribution, no pair-wise key sharing, and no complicated one-way keychain scheme. This paper contributes three concepts. First, they observed the fact that a sensor only communicates with a small portion of its neighbors and utilized it to reduce the overhead of key management. Second, they designed an effective key management scheme for HSNs by taking advantage of powerful H-sensors. Third, they utilized a public key algorithm - ECC for efficient key establishment among sensor nodes.

Zhou [15] proposed a key management method for heterogeneous wireless sensor networks based on ECC and *t-degree* trivariate symmetric polynomial. Provides node capture resiliency but rekeying consumes too much energy. A hybrid technique of Elliptic Curve Cryptography and symmetric key cryptography is combined. Symmetric keys are generated by a polynomial of sensor nodes to establish a secure communication link with their neighbors. Elliptic Curve Cryptography is used for message authentication and digital signature. Time Slice mechanism is introduced to regularly update session keys between nodes. The contribution of their proposed scheme is three folds: 1) Designed an effective key management scheme for HSNs by taking advantage of powerful H-sensors; 2) Utilization of Time Slice mechanism effectively prevented the cipher text analysis of adversary; 3) Designed a dynamic key update technique based on one-way hash chain and *t-degree* trivariate symmetric polynomial which significantly reduced the communication overhead in key agreement and update phase. In this proposed work, the researchers used the Elliptic Curve Cryptography for message authentication and digital signature. Symmetric keys are generated by a polynomial of sensor nodes to establish a secure link with their neighbors.

Zhang & Ji [16] establishes efficient and hybrid key management in wireless sensor networks (WSN) which is a challenging problem for the constrained energy, memory, and computational capabilities of the sensor nodes. Previous research on sensor network security mainly considers homogeneous sensor networks with symmetric key cryptography. Recent researches have shown that using asymmetric key cryptography in heterogeneous sensor networks (HSN) can improve network performance, such as connectivity, resilience, etc. Considering the advantages and disadvantages of symmetric key cryptography and asymmetric key cryptography, the paper propose an efficient and hybrid key management method for heterogeneous wireless sensor network ,cluster heads and base stations use public key encryption method based on elliptic curve cryptography (ECC), while using symmetric encryption method between adjacent nodes in the cluster.

The network model consists of two types of sensor nodes: Large number of low-end sensors (L-sensor) that has a lower energy and small number of high-end sensors (H-sensor) which have a great energy than L-sensor. There are powerful H-sensors serving as cluster heads in the EHKM method. Assume that the Base Station (BS) is trusted and has sufficient energy. All of nodes are static, and they know their own location information and H-sensors are uniformly deployed in the monitoring area. Due to energy constraints, L-sensors are not equipped with tamper-resistant hardware. H-sensors have larger communication bandwidth, computational capacity and larger storage spaces compared with L-sensors and are equipped with tamper-resistant hardware. Figure 1 explains the keys used in the method.

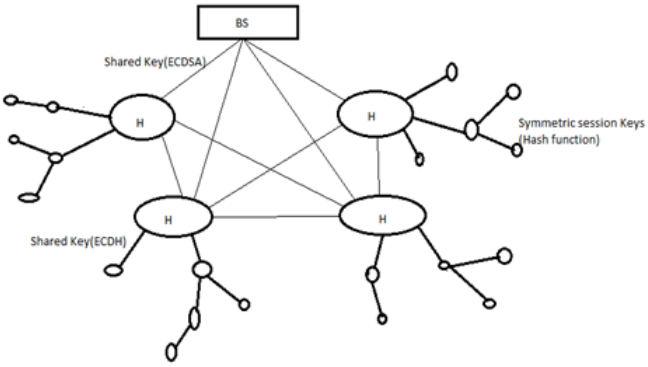

Figure 1: Heterogeneous network showing keys involved

This method needs relatively larger energy consumption, but its security is better, and cluster heads and the base station have enough energy to suppose the algorithm. The ECDH key exchange algorithm is adopted to establish a shared symmetric key which is used for communicating between cluster head and nodes in its cluster, between energy limited nodes in the cluster adopts one-way hash function to establish a session key. And it's effective in reducing the energy consumption within the cluster nodes. The proposed method can provide better security, scalability, connectivity and it can save storage space.

3. Conclusion

Zhanget al [16] proposed a combination of symmetric and asymmetric keys for communication and hence is called the highly efficient and hybrid. It also uses the H and L sensors with varying energy capacities. The ECDSA based on server used provides authentication. Symmetric keys generated by hash function are used to communicate between L sensors and the shared key obtained by using ECDH helps in communicating between H and L sensor. Thus it uses a hybrid combination of symmetric and asymmetric keys making it highly efficient. Addition and deletion of nodes is possible. This method can provide better security, scalability, connectivity and it can save storage space. Communication between cluster heads and base station can establish a secure link through signature encryption algorithm. This method needs relatively larger energy consumption. In future security can be enhanced by trying with other algorithms.

References

[1] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Compuer .Networks. 52 (12) (2008) 2292–2330,

[2] A.D. Wood, J.A. Stankovic, G. Zhou, DEEJAM: defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks, in: 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad-Hoc Communications and Networks, 2007, pp. 60–69.

[3] A. Hadjidj, M. Souil, A. Bouabdallah, Y. Challal, H. Owen, Wireless sensor networks for rehabilitation applications: challenges and opportunities, J. Networks and Computer Applications. 36 (2013) 1–15.

[4] K. Romer, F. Mattern, The design space of wireless sensor networks, Wireless Communication, IEEE 11 (2004) 54–61.

[5] A.K. Das, An unconditionally secure key management scheme for large-scale heterogeneous wireless sensor networks, in: Proceedings of the First International Conference on Communication Systems and networks, IEEE Press, Bangalore, India, 2009, pp. 653–662.

[6] M. Boujelben, H. Youssef, R. Mzid, M. Abid, IKM—an identity based key management scheme for heterogeneous sensor networks, Journal on Communications 6 (2) (2011) 185–197.

[7] X. Du, Y. Xiao, and Guizani. An effective key management scheme for heterogeneous sensor networks. Ad Hoc Networks, 5(1):24–34, 2007.

[8] Eschenauer L, Gligor V D. A key-management scheme for distributed sensor networks, Proceedings of the 9th ACM conference on Computer and communications security.ACM, 41-47, 2002.

[9] Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks, 2003 Symposium on Security and Privacy. IEEE, pp. 197-213, 2003.

[10] Azarderakhsh R, Reyhani-Masoleh A, Abid Z E. A key management scheme for cluster based wireless sensor networks, EUC'08. IEEE/IFIP International Conference on Embedded and Ubiquitous Computing. IEEE, Vol.2, 222-227, 2008.

[11] G. Gupta and M. Younis. Load-balanced clustering of wireless sensor networks. In Proceedings of IEEE International Conference on Communications, ICC’03, 2003.

[12] X. Du and Y. Xiao. Energy efficient chessboard clustering and routing in heterogeneous sensor networks. Journal of Wireless and Mobile Computing, 1(2):121–130, 2006.

[13] A. Liu, P. Kampanakis, and P. Ning.Tinyecc: Elliptic curve cryptography for sensor networks (version 0.3), 2007.

[14] Du X, Guizani M, Xiao Y, et al. Transactions papers a routing-driven Elliptic Curve Cryptography based key management scheme for Heterogeneous Sensor Networks. Wireless Communications, IEEE Transactions on, Vol.8, No.3, 1223-1229, and 2009.

[15] Zhou R, Yang H. A hybrid key management scheme for Heterogeneous wireless sensor networks based on ECC and trivariate symmetric polynomial, 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering (URKE). IEEE, Vol.1, 251-255, 2011.

[16] Zhang Ying, JiPengfei, an Efficient and Hybrid Key Management for Heterogeneous Wireless Sensor Networks, 26th Chinese Control and Decision Conference (CCDC) IEEE, 1881-1885, 2014