Comparison between LEAP and Random key pre-distribution scheme in wireless sensor network

Bakhshi Larmaei Reza,
Department Of computer Islamic Azad University,Ayatollah Amoli branch Amol,Iran

Reza0642@gmail.com

Abstract-Wireless sensor networks use in a wide range of applications from military to civilian. A wireless sensor network is made from a large number of sensor nodes. The limited battery power of sensor nodes, the poor data processing capabilities and the short range of radio are the main characteristics of WSN. More importantly, sensor nodes are often distributed randomly on specific areas and work environments unattended. They are exposed to all kinds of attacks, so security is the first concern of it. In order to keep communications secure, data must be encrypted and authentication are important. Therefore, a prerequisite for authentication and encryption key management should be considered carefully. So far, several key management methods have been introduced for Sensor Networks, but due to the large variety of appliances and different security needs, They can not provide a unique key management for all applications considered. As a result, the key management strategy for each application is so important. In this paper, Several key management scheme in wireless sensor networks is studied. The proposed key management schemes are compared with each other.

Keywords: **Wireless Sensor Networks Security Key Management**

I. Introduction

For safe keeping communication between sensor nodes, important data should be encrypted and authenticated. Therefore, key management, encryption and authentication is a prerequisite, it is important[1]. Among all key management mechanisms in wireless sensor networks, key pre-distribution mechanism provides a good balance between overhead storage and processing power and as the most appropriate mechanisms for wireless sensor networks is studied.

II. Random key predistribution scheme

Eschenauer and Gligor first proposed a random key predistribution scheme [11]. In the remainder of this paper, we refer to their approach as the basic scheme. Let m denote the number of distinct cryptographic keys that can be stored on a sensor node. The basic scheme works as folws.

Before sensor nodes are deployed, an initialization phase is performed. In the initialization phase, the basic scheme picks a random pool (set) of keys S out of the total possible key space. For each node, m keys are randomly selected from the key pool S 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, [s], is chosen such that two random subsets of size m in S will share at least one key with some probability p. After the sensor nodes are deployed, a key-setup phase performed. The nodes first perform key-discovery to find out with which of their neighbors they share a 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 in their key rings can then verify that their neighbor actually holds the key through a challenge response protocol. The shared key then becomes the key for that link.

ISSN 2519-0814

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 whom they did not happen to share keys with in 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.

In key setup phase, a large key-pool of KP keys and their identities are generated. For each sensor[2], k keys are randomly drawn from the key-pool KP without replacement. These k keys and their identities form the key-chain for a sensor node. Thus, probability of key share among two sensor nodes becomes $p = ((KP-k)!)^2/((KP-2k)!KP!)$. In shared-key discovery phase, two neighbor nodes exchange and compare list of identities of keys in their key-chains. Basically, each sensor node broadcasts one message, and receives one message from each node within its radio range where messages carry key ID list of size k. Cluster key grouping scheme [Hwang et al. 2004] proposes to divide key-chains into C clusters where each cluster has a start key ID. Remaining key IDs within the cluster are implicitly known from the start key ID. Thus, only start key IDs for clusters are broadcasted during shared-key discovery phase which means messages carry key ID list of size c instead of k. Another solution is given by Pair-wise key establishment protocol [3]which requires every sensor node to have a unique ID which is used as a seed to a PRF. Key IDs for the keys in the key-chain of node S are generated by $PRF(ID_A)$. Thus, broadcast messages carry only one key ID. Also, storage, which is required to buffer received broadcast message before processing, decreases substantially. But, a sensor node has to execute PRF(ID) for each broadcast message received from a neighbor. Transmission range adjustment scheme [Hwang and Kim 2004] proposes sensor nodes to increase their transmission ranges during shared-key discovery phase. Nodes return to their original optimal transmission range once the keys are discovered. Idea is to decrease communication burden in path-key establishment phase, and to save energy while still providing a good key connectivity.

It is possible to protect key identities broadcasted in shared-key discovery by using a method similar to Merkle Puzzle [Merkle 1978] which substantially increases processing and communication usage. After shared-key discovery phase, some node pairs may not be able to find a key in common. These pairs apply path-key establishment phase to communicate securely through other nodes. Scalability and resilience of the solutions can be improved by using larger key pools[4]. But, larger key-pool means smaller probability of key share because key-chain size may not increase due to storage limitations. Probability that a link is compromised, when a sensor node is captured, is k/KP which is very high for small key-pools, and produces low resilience.

III. Leap time base key management scheme

LEAP provides multiple keying mechanisms for providing confidentiality and authentication in sensor networks. We first motivate and present an overview of the different keying mechanisms in the following Section before describing the schemes used by LEAP for establishing these keys[6].

The packets exchanged by nodes in a sensor network can be classified into several categories based on different criteria, e.g. control packets vs data packets, broadcast packets vs unicast packets, queries or commands vs sensor readings, etc. The security requirements for a packet will typically depend on the category it falls in. Authentication is required for all types of packets, whereas confidentiality may only be required for some types of packets. For example, routing control information usually does not require confidentiality, whereas (aggregated) readings reported by a sensor node and the queries sent by the base station may require confidentiality[7]. We argue that no single keying mechanism is appropriate for all the secure communications that are needed in sensor networks. As such, LEAP supports the establishment of four types of keys for each sensor node – an individual key shared with the base station, a pairwise key shared with another

ISSN 2519-0814

sensor node, a cluster key shared with multiple neighboring nodes, and a group key that is shared by all the nodes in the network.

IV. Conclusion

In this paper 2 main methods of providing security in the wireless sensor network are evaluated. Leap has a better performance than random pre-distribution scheme In relation to the key bindings and overhead storage.

References

[1] Wang, L., Khan, S., et al.: Energy-aware parallel task scheduling in a cluster. Future Generation Computer Systems 29(7), 1661–1670 (2013)

[2] Luo, X., Xu, Z., Yu, J., Chen, X.: Building Association Link Network for Semantic Link on Web Resources. IEEE Transactions on Automation Science and Engineering 8(3), 482–494 (2011)

[3] Xu, Z., Luo, X., Yu, J., Xu, W.: Measuring semantic similarity between words by removing noise. Concurrency and Computation: Practice and Experience 23(18), 2496–2510 (2011)

[4] Yuan, D., Yang, Y., Liu, X., Li, W., Cui, L., Xu, M., Chen, J.: A highly practical approach towards achieving minimum datasets storage cost in the cloud. IEEE Transactions on Parallel and Distributed Systems 24(6), 1234–1244 (2013)

[5] Zhang, X., Liu, C., Nepal, S., Pandev, S., Chen, J.: A privacy leakage upper-bound constraint based approach for cost-effective privacy preserving of intermediate datasets in cloud. IEEE Transactions on Parallel and Distributed Systems 24(6), 1192–1202 (2013)

[6] Wang, L., Tao, J., et al.: G-Hadoop: MapReduce across distributed data centers for data-intensive computing. Future Generation Computer Systems 29(3), 739–750 (2013)

[7] J. Spencer. The Strange Logic of Random Graphs. Number 22 in Algorithms and Combinatorics. 2000.

[8] David W. Carman, Peter S. Kruus, and Brian J. Matt. Constraints and approaches for distributed sensor network security. NAI Labs Technical Report #00-010, September 2000.

ISSN 2519-0814