Location Aware Pair-wise Key Generation Schemes for Wireless Sensor Networks

Debargh Acharya and Vijay Kumar
SCE, Computer Science
University of Missouri-Kansas City
5100 Rockhill Road
Kansas City, MO 64110, USA
dabn7; kumarv{@umkc.edu}

Abstract: Secure communication among large numbers of randomly scattered wireless sensor nodes usually requires pair-wise keys. In all existing schemes of secured communication pair wise keys are generated and distributed to nodes wishing to communicate. The key generation phase is usually well-secured but the key distribution is vulnerable to security threats. In this work, we investigate the key distribution problem in large wireless sensor networks and present one of two secure communication schemes we developed. Unlike others, our schemes do not store a key chain in the memory from a universal key space and eliminate key broadcast. We have made the key generation phase relatively more secured with the use of location information. Authentication of sensor nodes is also an important issue and has been taken into consideration in our schemes. Simulation of our schemes illustrate that they outperform some existing key schemes and incurs less transmission and storage cost.

Keywords: Wireless sensor networks; key distribution; location parameters; security.

Introduction

In this paper we present secured communication scheme in Wireless Sensor Network (WSN) [1, 2]. The contributions of this paper are:

a. The use of geographical location as one of the parameters to strengthen our security scheme.
b. A way to avoid key-distribution and to reduce the cost of securing the network.
c. A comparison of their performance with some existing schemes to illustrate the strength of our schemes.

The idea of key broadcast [7, 8, 9] seems useful; however, in reality it is fairly unreliable [14], especially for devices with IEEE 802.15.4 radio packets, a de facto standard in WSN. While the maximum broadcast packet size is a few kilobytes of payload, an individual 802.15.4 radio packet only carries about few hundred bytes of data and this mode is inherently unreliable because the list of recipients is unknown. Hence, schemes that transfer packets with subset of keys may lose a portion of data as they would be fragmented during broadcast. For these reasons we took a different approach for generating and using the keys for communication instead of distributing them to nodes wishing to communicate.

Our Security Schemes: Several works have discussed the problem of devising a secure mechanism for key generation and distribution [4, 5, 6, 7, 8, 9, 10, 11, 12]. Authentication of sensor node is also a major area of research while developing schemes for key generation and distribution. Our analysis of existing schemes identifies key-distribution portion as one of the most time consuming activities which can be avoided altogether for establishing inter-node communication. Thus, our objective is to develop cost-effective and reliable communication schemes by avoiding key broadcast to eliminate a certain portion of risk involved in transmission loss (leading to redistribution of keys and more consumption of power).

For our work we use a WSN and divide it into clusters. Each cluster has a head node (*H*) which communicates wirelessly with its nodes. We have used the location (Lon-L/Lat-L) to develop our scheme referred to as Location-dependent Hash (LH) chain based scheme. To the best of our knowledge L/L has not been used in any earlier works.

LH Scheme: It uses (L/L) of sensor nodes to generate the individual key sets which are then used to generate the common pair-wise key for any two nodes wishing to communicate. This

approach makes sure that no two nodes can have any keys in common. The pair-wise key generation scheme has two steps: (a) deployment of sensors and individual key generation by nodes using (L/L) and (b) authentication of sensor nodes and generation of pair wise key. Figure 2 shows our scheme of using location information in hash chains to generate individual key rings. H generates its key using hash function *h* and passes it to the next node of the cluster. The key generation continues until the last node of the cluster has generated its key. In our scheme we require three keys in the key ring set of each node. Since key generation is hash chain based, only H has the ability to generate individual keys for any node in its cluster as it knows the location of all the nodes in its cluster. The H uses this concept later in the pair wise key generation mechanism.

Hash chains are lightweight crypto elements and are suitable for applications in WSN.

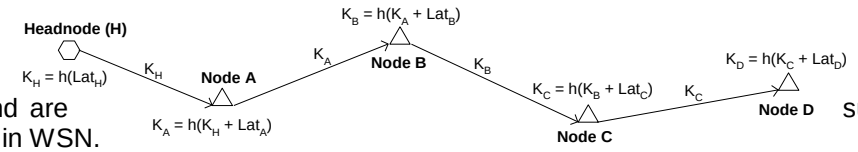


Figure 2. Hashing of Location Information for individual key generation

Authentication of sensor nodes and generation of pair wise key: We consider two nodes *A* and *B* with identities ID_A and ID_B respectively in the same cluster and show how pairwise keys for them are generated (Figure 3). Suppose *A* has K_{A1}, K_{A2} and K_{A3} and B has K_{B1}, K_{B2} and K_{B3}. These individual keys have been generated by the three hash chains discussed above.

If *A* wants to communicate with *B*, then the pair wise key generation starts by *A* sending its ID_A and an encrypted message E_{KA1} contains 3 entities: (a) two of the three individual keys of *A* chosen randomly and (b) the identifier ID_B of *B*. E_{KA1} is a message encrypted using the individual key of *A* (e.g., K_{A1}), this key, K_{A1}, is not and hence included in the message. *A* sends this encrypted message E_{KA1} to *H* requesting a common pairwise key, K_{AB}, for communication.

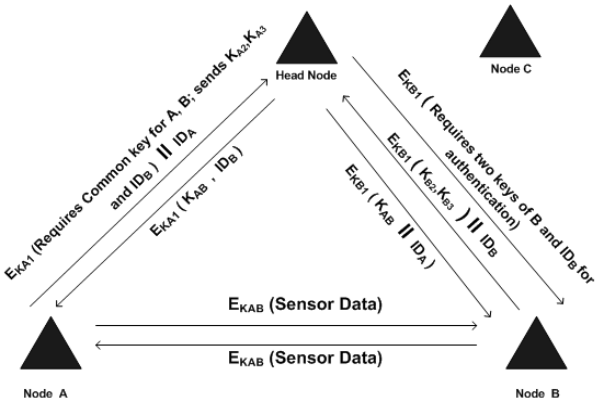


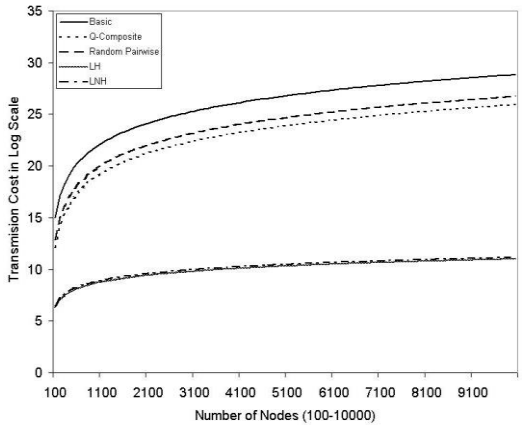
Figure 3. LH Scheme.

H checks the authenticity of these two nodes before it generates the pairwise key for *A* and *B*. *H* checks the identifier of *A* and initially assumes that *A* is a local node of the cluster. Since all the keys of *A* were earlier generated by using a hash chain starting from *H*, it can re-hash multiple times and calculate all the three keys of *A*. It decrypts E_{KA1} (by applying the three keys of *A* using trial and error method) and get two keys of *A* in the message. *H* authenticates *A* if two of the three generated keys match with the key pair in the message. *H* also knows that *A* wants to communicate with *B* (the encrypted message sent by *A* to *H* includes the identifier of the node it wants to communicate with) so it generates the individual keys of *B* using hash chain and stores it into its memory together with the individual keys of *A*. *H* authenticates node *B* next.

After a successful authentication of *A* and *B* the common pairwise key generation starts. *H* randomly selects the individual keys of *A* and *B* (say K_{A1} K_{A2} K_{B2} K_{B3}) and generate $K_{AB} = K_{A1} \oplus K_{A2} \oplus K_{B2} \oplus K_{B3}$ (Common key for *A* & *B* generated by XORing). *H* sends K_{AB} and ID_B to node *A* by encrypting it with K_{A1} and K_{AB} and ID_A to node *B* by encrypting it with K_{B1}. Nodes *A* and *B* will decrypt this common pairwise key and ID information by using their respective keys K_{A1} and K_{B1}. Node *A* understands that this pairwise key is to communicate with a node in its cluster with identifier ID_B. Similarly, node *B* understands that this pairwise key is to communicate with a node in its cluster with identifier ID_A. The algorithm for pair wise key generation is summarized below:

a. *A* sends its ID_A and an encrypted message E_{KA1} (containing two randomly chosen keys of *A* and ID_B) to *H* requesting a pairwise key $K_{AB.}$ The message is encrypted using one of the keys of *A*, say K_{A1}.
b. *H* decrypts the message using key K_{A1} (it can generate all the three keys of *A* using hash chain) and verifies *A* is an authenticate node. It also generates the three individual keys of *B*.
c. *H* sends an encrypted message E_{KB1} to *B* requesting its ID and the other two keys of *B*.
d. *B* decrypts this message E_{KB1} using K_{B1} and sends back an encrypted message E_{KB1} (containing ID_B and two other keys of *B* say K_{B2} and K_{B3}). The message is encrypted using one of the keys of B, say K_{B1}.
e. *H* decrypts this message using key K_{B1} and authenticates *B*.
f. *H* generates common pairwise key K_{AB} by XORing the randomly chosen keys of *A* and *B*. ($K_{AB} = K_{A1} \oplus K_{A2} \oplus K_{B2} \oplus K_{B3}$).
g. *H* sends the common pairwise key K_{AB} and ID_B to *A* by encrypting it with K_{A1}.
h. *H* sends the common pairwise key K_{AB} and ID_A to *B* by encrypting it with K_{B1}.

A and *B* may now communicate with each other using the common symmetric key. Figures 5 and 6 records our result. Note that the graph also includes the performance of our second scheme referred to as LNH.



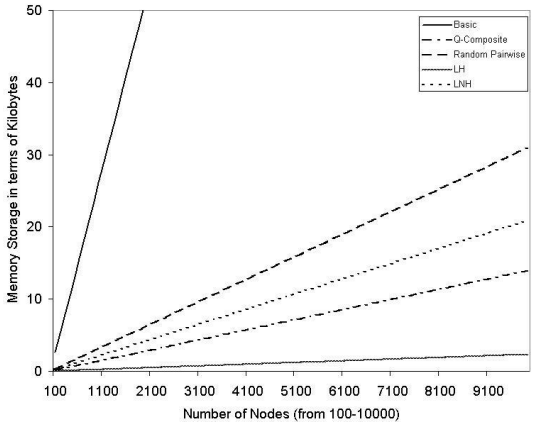


Fig. 5 and 6. Data Transmission and storage utilization comparison in all the Schemes.

Conclusion

In this work we have presented one of the two unicast based pairwise key generation and distribution schemes. A prominent feature in the key generation process in the LH scheme includes generation of one way hash key chains. Broadcast of key ring sets have been avoided (due to the inherent feature of unreliability of data broadcast in sensor nodes) unlike other schemes. This idea has led to significant reduction of cost in terms of transmission and storage. We compared our LH scheme with other schemes (Basic scheme, Random pairwise scheme and Q-Composite scheme, which are broadcast based) and evaluated how its performance in terms of total data transmission and memory storage in a cluster. Simulation and analysis shows LH scheme significantly outperforms all the broadcast based schemes in both the parameters. Thus, we can safely argue that unicast based schemes perform better than conventional broadcast based schemes during pairwise key generation and distribution in wireless sensor networks.

References

[1] A. Bharathidasan, and V. Ponduru, “Sensor Networks: An Overview”, IEEE Infocom, Hongkong, 7-11 March 2004.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A Survey on Sensor Networks”, IEEE Communications Magazine, August 2002.

[3] A. Perrig, J. Stankovic, and D. Wagner, “Security in Wireless Sensor Networks”, Communications of the ACM, June 2004.

[4] C. Haowen, and A. Perrig, “Security and Privacy in Sensor Networks”, IEEE Computer Society, October 2003.

[5] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, “SPINS: Security Protocols for Sensor Networks”, IEEE Infocom, Anchorage (Alaska), 22-26 April 2001.

[6] S.A. Camtepe, and B. Yener, “Key Distribution Mechanisms for Wireless Sensor Networks: A Survey”, Technical Report TR-05-07 Rensselaer Polytechnic Institute, Computer Science Department, March 2005.

[7] L. Eschenauer, and V. D. Gligor, “A Key-Management Scheme for Distributed Sensor Networks”, In 9th ACM Conference on Computer and Communications Security, Washington DC, 18-22 November 2002.

[8] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks”, In IEEE Symposium on Research in Security and Privacy, California, 11-14 May 2003.

[9] D. Liu, and P. Ning, “Location-Based Pairwise Key Establishment for Static Sensor Networks”, In 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, Fairfax, 31 October 2003.

[10] W. Du, J. Deng, Y. Han, and P. Varshney, “A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks” Proc. 10th ACM Conf. Computer and Comm. Security (CCS), Washington DC, 27-30 October 2003.

[11] D. Liu, and P. Ning, “Establishing Pairwise Keys in Distributed Sensor Networks”, In 10th ACM Conference on Computer and Communications Security CCS, Washington DC, 27-30 October 2003.

[12] S. Zhu, S. Setia, and S. Jajodia, “Leap: Efficient security mechanisms for large-scale distributed sensor networks”, In 10th ACM Conference on Computer and Communications Security (CCS), Washington DC 27-30 October 2003.

[13] A. Perrig, R. Canetti, D. Song, and D. Tygar, “Efficient authentication and signing of multicast streams over lossy channels”, In Proceedings of the IEEE Symposium on Security, May 2000.

[14] O. Goldreich, S. Goldwasser, and S. Micali, “How to construct random functions” Journal of the ACM 33, 4 (October), 792–807.

[15] D. Liu, and P. Ning, “Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks”, In Proceedings of the 10th Annual Network and Distributed System Security Symposium, February 2003.

[16] D. Liu and P. Ning, “Multi-level mTESLA: Broadcast authentication for distributed sensor networks,” ACM Transactions in Embedded Computing Systems (TECS), August 2004.

[17] http://www.sunspotworld.com.

[18] www.sunspotworld.com/docs/Purple/SunSPOT-OwnersManual.pdf

[19] N. Aakvaag, M. Mathiesen and G. Thonet, “Timing and Power Issues in Wireless Sensor Networks,” An Industrial Test Case, In Proceedings of the 2005 International Conference on Parallel Processing Workshops, 2005.