A KEY MANAGEMENT APPROACH FOR WIRELESS SENSOR NETWORKS

Ali Bagherinia, Akbar Bemana, Sohrab Hojjatkhah, Ali Jouharpour

Department of Computer Engineering, Islamic Azad University-Dehdasht Branch, Dehdasht, Iran

ABSTRACT

In this paper we presenta key management approach for wireless sensor networks. This approach facilitating an efficient scalable post-distribution key establishment that provides different security services. We have developed and tested this approach under TinyOs. Result shows that this approach provides acceptable resistance against node capture attacks and replay attacks. The provision of security services is completely transparent to the user of the WSNs. Furthermore, being highly scalable and lightweight, this approach is appropriate to be used in a wireless sensor network of hundreds of nodes.

KEYWORDS

Sensor networks,key management , scalability , flexibility , resistant.

1. INTRODUCTION

Recent advances in electronic and computer technologies have paved the way for the proliferation of wireless sensor networks (WSN) [1,2]. Sensor networks usually consist of a large number of ultra-small autonomous devices. Each device, called a sensor node. Each sensor node measures necessary parameters from round area and communicate it's with radio sender through electrical signal. Processing of this signal extracts specification such as object placement or around events. Figure 1 shows modular structure of each multi sensing sensor node. Each sensor node consists of: multi sensing interface and A/D (for sensing corresponding analog area such as pressure, temperature …), memory, CPU, RF and controller [3].

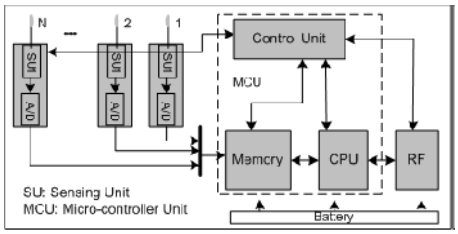

Figure 1. Sensor node structure with multiple sensing units

DOI : 10.5121/ijitmc.2014.2301

This key agreement problem is a part of the key management problem, which has been widely studied in general network environments. There are three types of general key agreement schemes: trusted- server scheme, self-enforcing scheme, and key pre-distribution scheme. The trusted server scheme depends on a trusted server for key agreement between nodes ,e.g., Kerberos[5].This type of scheme is not suitable for sensor networks because there is usually not rusted infrastructure in sensor networks. The self-enforcing scheme depends on a symmetric cryptography, such as key a agreement using public key certificates. However, limited computation and energy resources of sensor nodes of ten make it undesirable to use public key algorithms, such as Daffier-Hell man key agreement[6]or RSA[7],as pointed outing[8].The third type of key agreement scheme is key redistribution, where key information is distribute among all sensor no despair to deployment. If we know which nodes are more likely to stay in the same neighborhood before deployment, key scan be decide dapriori. However, because of the randomness of the deployment, knowing the set of neighbors deterministically might not be feasible.

.

2. Related work

Key management is an essential challenge in a large-scale and resource-limited particularly WSNS. In [28],[11], [12], [13], [14], [15], [16], [17] a number of pair-wise symmetric key establishment schemes have been recently proposed. Most of them use the idea of probabilistic key sharing [14] to establish trust between two nodes, each with different emphasis on enhanced security protection [11], flexibility of security requirements [17], high probability of key establishment and reduced overhead [15], or utilization of deployment knowledge [12]. Such pairwise keys can be used to authenticate a node's identity or messages; how- ever, they cannot handle the fabricated sensing data injected by compromised nodes. Instead, semantic verification of the data is required to detect the fabricated ones. Secure Diffusion exploits location-based key management to achieve this goal. Because the data authentication keys are bound to geographic locations, the compromised nodes outside the targeted region, no matter how many there are, cannot fabricate sensing data without being detected.

Secure routing has been extensively studied in the context of ad-hoc networks [18], [19], [20], [21]. However, none of these protocols can be applied in sensor networks, because none addresses the unique feature of data-centric communication, and the network scale is limited by the excessive number of keys each node should store. The challenges of secure sensor routing are discussed in [22], together with security threat and counter-measurement analysis on a few popular routing protocols. However, it does not consider the fabricated data injection attacks launched by compromised nodes.

Two recent studies of SEF [23] and Hop-by-Hop Authentication [24] address the problem of filtering the fabricated data en-route in sensor networks. Such early drop of malicious traffic can potentially save precious energy resources at forwarding nodes. Secure Diffusion takes a different approach that quarantines the malicious traffic through implicit rate control and negative reinforcement mechanisms. As a result, Secure Diffusion is resilient to an increasing number of compromised nodes, whereas both SEF and Hop-by-Hop Authentication completely lose security protection when the attacker has compromised beyond a small, fixed number of nodes.
There are a few recent security proposals that explicitly involve the geographic locations. The Echo protocol [25] exploits an on-site verifier node with ultrasound transceiver to verify a location claim. A recent secure routing proposal TRANS [26] monitors the behavior of static

sensor nodes, and then bypasses the areas of misbehaving nodes in the route. The pair wise key establishment scheme in [16] exploits a location- aware deployment model and pre-distributes pair wise keys between nodes that are expected to be close to each other. However, Secure Diffusion differs from all these work in that it binds keys to locations, and provides a scalable secure data dissemination protocol for sensor networks.

Thee exist a number of key pre-distribution schemes. A naive solution isotope tall the nodes carry a Master secret key. Any pair of nodes can use this global master secret key to achieve key agreement and obtain a new pair wise key. This scheme does not exhibit desirable network resilience: if one node Is compromised, the security of the entire sensor network will be compromised. Some existing studies Suggest storing the master key in tamper- resistant hardware to reduce the risk, but this increases the Cost and energy consumption of each sensor. Furthermore, tamper- resistant hardware might not always be safe[9]. Another key pre-distribution scheme isolate each sensor carry N-1 secret pair Wise keys, each of which is known only to this sensor and one of the other N-1 sensors(assuming Nis the total number of sensors).

The resilience of this scheme is perfect because compromising one Node does not affect the security of communications among other nodes; however, this scheme is impractical for sensor switch an extremely limited amount of memory because N could be large. Moreover, adding new nodes to a pre-existing sensor network is difficult because the exist in nodes do not have the new nodes 'keys. Because of their small size, limited processing power, and unattended deployment, individual sensor nodes are highly prone to security compromises.

Therefore, it is important to build security in to the network architecture and protocols, so that a sensor network can successfully operate in the presence of both component fail rues and malicious attacks [10]. This paper consists of: related work (section 2), proposed approach (section 3), simulation (section 4), results and conclusion.

3. Proposed approach

In this section we describe our key management approach. Our approach is a post-deployment key management scheme which deal scalability and flexibility issues and is resistant to node capture attacks.

All of the direct communications in wireless sensor networks can be divided into the two types of one-to-one and one-to-many. To secure these communication sour key establishment approach establishes the following kinds of keys:

i. *Pair-wise(PW) key* that is established between two neighbors to protect their for one-to-one communications.
ii. *Broadcast(BC)key* that is established in order to secure the broad cast messages sent by a node to its neighbors.
iii. *Node-zase(NB)key* that is established in order to secure the communication between a node and the base station (note that this communication is not necessarily direct). A message encrypted by this key, can only be decrypted by the base station.

Since the *pair-wise* and *broad cast keys* are essentially established among neighboring nodes , the first phase of key establishment is *neighbor discovery*. This is achieved in two steps by a pair

of hand shake messages. In the first step, node*s* broad casts a specific type of message containing its ID, so that every other node in *s*'s communication range (like *r* for example) can receive it. Were fero this message as a ping *message*. Every node receiving the ping message answers back to the sender(*s*) with a *pong message* containing its ID (steps1 and 2 in Figure 2) .Node*s* can then add *r* to its own neighbor list. After a sufficient amount of time (see Table IV and more explanations in Section IV -B), *s* will discover all of its neighbors and this phase will be finished.

When the neighbor discovery phase is over, node *s* computes its own node-base key and its pair-wise keys with its neighbors as well as their broad cast keys as follows:

$$Nib_S = Func(s||base\ Station\ Address||K)$$

$$PS_{s,r} = F(min(s,r)||max(s,r)||GMK)$$

$$BS_S = Func(s||GMK)$$

where"||"is the concatenation operator and *Func* is a secure pseudo-random function usually implemented by a hash function such MD5. GMK is a global master key that is distributed to all nodes before deployment of the network. As we will explain later, *GMK* will eventually be deleted from the memory of the nodes in order to make the approach more secure against node capture attacks.

Step	Message
1	$s \rightarrow r: \{s\}$
2	$r \rightarrow s: \{r\}$
3	$s \rightarrow r: \{s, PS_{sr}, NIB\ A\} NIBB_s$
4	$s \rightarrow r: \{s, BS_r, NB\} NibB_s$
5	$s \rightarrow r: \{r, Nib_A, Nib\ B\} PS_{sr}$
6	

Figure 2. STEPSOFKEYESTABLISHMENTPROTOCOL

Whenthesecalculationsareover,node*s*hasacompletetableofrelatedkeys.However,node*r*'skeytableis notquitecompleteasitdoesnothaveanyentrycorrespondingtonode*s*.Thus,node*s*hastosendamessage *M1*containingthesekeystonode*r*.Obviously,*M1*shouldnotbesentinplain.Therefore,node*s*shouldcal culateanappropriatekeytoencrypt*M1*withitandthensendtheencryptedversionof*M1*tonode*r*.Aproper key,aswewillsee,isthenode-basekeyofnode*r*whichcanbefollowedby*s*asfollows:

$$NibB_r = Func(r \,||\, baseStationAddress \,||\, GMK)$$

Having this key, node *s* can encrypt and send to *r* the key it shares with it as well as its own broad cast key. The related messages are the following (Steps3 and 4 in Figure 2):

$$s \rightarrow r: \{s, PS_{sr}, NIB\ A\} NIBB_s$$

$$s \rightarrow r: \{s, BS_r, NB\} NibB_s$$

where Nib_A and Nib_B are two non cesto guarantee the freshness of these messages.

After sending these two messages, node *s* will delete the node-base key of node *r* from its memory. Therefore the only non-base station node that can decrypt these message s is node *r* (note that we assume the base station is secure). Node *s* will also delete the master key GM*K* from its memory.

Upon receiving the keys, node r will answer back to node s by sending a message containing the non ces Nib_A and Nib_B. This message is encrypted with the pair-wise key of s and r (Sstep5 in Figure 2). At this point, key establishment is complete.

Notice how this message exchange enforces the *scalability* aspect of our protocol: related keys can be established when a new node is added to a previously deployed network. Any new node that joins the network (such as s) can initiate the key establishment phase by broad casting a *ping message*. Following that, related keys are calculated by then ew node. Then the broad cast keys of this added node, as well as its pair-wise keys with each of its neighbors are sent to related neighbors, encrypted with their node-base keys. Note that using the node-base keys for this purpose is quite an appropriate choice in order to make the protocol scalable and secure. This is because the already available network nodes have already deleted the master key GMK from their memory and consequently cannot use it to either calculate the keys orde cryptany message encrypted with it. It is not a good idea touse the broad cast key of previously joined neighbor nodes (similar to r) since other neighbors of r have that key available and can decrypt messages encrypted with it; a fact that results in providing a looser security scheme.

The deletion of master key GMK and the temporarily calculated node-base key of r by s as mentioned above, makes the protocol resilient to node capture attacks by reducing the effects of capturing a node to its neighborhood and *not the entire network*. Since the needed time for key establishment is negligible, we can assume that the adversary does not have enough time to find the master key GMK before it is deleted from the memory of the nodes (see also LEAP [4] for a similar assumption). On the other hand, newly joined nodes must come with the master key GMK in order to calculate the cryptographic keys. Therefore, the adversary cannot gain any use ful information by introducing new nodes to the network as a result of not having access to GMK. In addition to that, it is important to note that if one of the above mentioned messages in key establishment protocol is not delivered, the receiving node will not get stuck. If node s does not receive the last message of the protocol (Step5 in TableIII), it will not add any entry for node r in its key table.

4. Simulation

Our key management approach is implemented in Tiny Os[27] which is an event-driven operating system commonly used on WSN nodes (motes). Results are shown in Table 1 and Figure 3.

Table 1. Required energy and time before deleting the glbal key

Phase	Neighbor discovery	Key computation	Key Sending
Energy (nJ)	1592640	157	38049000
Time (ms)	1000	10	10

Our key establishment approach is 10 bytes, which provides strong security (2^{80} bit key space) fo r sensor network applications. As a result, I kna very dense network where $d = 50$ will have $M \approx 1KB$. Although this value of d is far more than enough to keep the network connected, this

memory over head is well within the memory capabilities of motes (MICA 2 motes have 4KB of RAM).

During the key establishment phase, prior to deletion of the master key ,and versary has a chancet of in d it and use it to derive all the other keys. However, this time is so small that probability of having a nad versary capture a mote during it is minimal. Table IV shows there lated duration t hat it takes to delete the master key from memory of a newly added mote during its initialization phase. These results are of simulations using an internal simulator coming with Tiny Os (Tossim).

The estimated amount of energy consumption for each phase of key establishment for the same network (d=50) is presented in Table1 as well. This estimation was performed by multiplying the total amount of communications by an average communications cost of 18 $\mu J/bit$).As a result, the estimated energy consumption of our key management scheme is approximately 0.4J comparing to PIKE-2D [28] that is more than 8J or PIKE-3D[28] which is around 6J. This high energy efficiency of our platform comes with a comparable cost in terms of memory over head; it uses about 1000bytes of memory to establish and manage the keys while PIKE-2D and PIKE-3D need around 600 bytes and 500 bytes respectively.

In our scheme the effects of having a node captured is reduced to its neighborhood, its broad cast key and its node-base key are only keys that can be discovered by the adversary. This is a small fraction of established keys and secure communication still remains possible in other parts of the network.

Enegy consumption according to number of malicious nodes is shown in Figure 3. It is clear that with larg number of malicious nodes consumption of energy is less than SEF and Hop-by-Hop Authen- tication approchs.

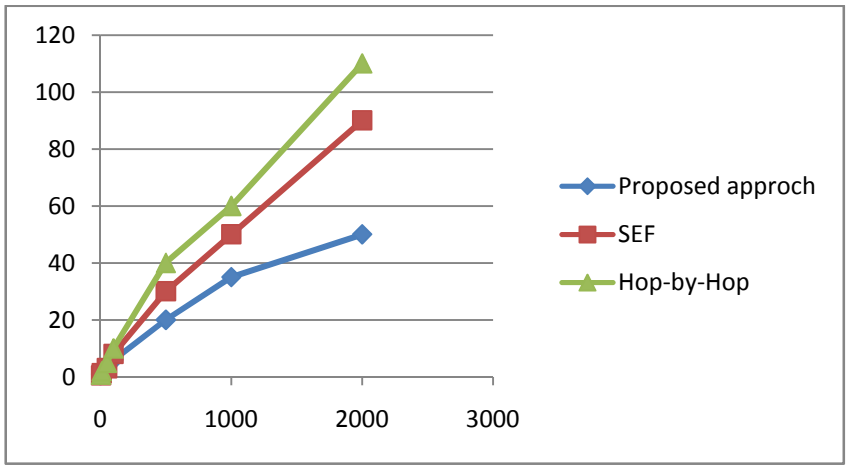

Figure 2 Enegy consumption according to number of malicious nodes.

5. conclusion

In this paper we introduced a post-distribution key management approach that provides several security services such as acceptable resistance against node capture attacks and replay attacks. It is allows for high scalability while being easy to use and transparent to the users and light weight. Simulation result shows that energy consumption in proposed approach with larg number of malicious nodes in contast to other approaches is less.

References

[1] 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.

[2]C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, pages 56-67, Boston, MA, Aug. 2000. ACM Press.

[3] Bagherinia, "OPTIMIZED TASK ALLOCATION IN SENSOR NETWORKS", IJITMC, Vol.1, No.3,August 2013.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, August 2002.

[5] B. C. Neuman and T. Tso, "Kerberos: An authentication service for computer networks," IEEE Communications, vol. 32, no. 9, pp. 33–38, September 1994.

[6] W. Diffie and M. E. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, vol. 22, pp. 644–654, November 1976.

[7] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.

[8] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. D. Tygar, "Spins: Security protocols for sensor networks," in Proceedings of the 7th Annual ACM/IEEE Internation Conference on Mobile Computing and Networking (MobiCom), Rome, Italy, July 2001, pp. 189–199.

[9] R. Anderson and M. Kuhn, "Tamper resistance - a cautionary note," in Proceedings of the Second Usenix Workshop on Electronic Commerce, November 1996, pp. 1–11.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: AScalable and Robust Communication Paradigm for Sensor Networks. InACM MOBICOM, 2000.

[11] H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. In IEEE Symposium on Security and Privacy, 2003.

[12] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney. A Key ManagementScheme for Wireless Sensor Networks Using Deployment Knowledge. In IEEE INFOCOM, 2004.

[13] W. Du, J. Deng, Y. Han, and P. Varshney. A Pairwise Key Predistribution Scheme for Wireless Sensor Networks. In ACM CCS, 2003.

[14] L. Eschenauer and V. Gligor. A Key-Management Scheme for Distributed Sensor Networks. In ACM CCS, 2002.

[15] D.LiuandP.Ning. EstablishingPairwiseKeysinDistributedSensorNetworks.InACMCCS,2003.

[16] D.LiuandP.Ning. Location-BasedPairwiseKeyEstablishmentsforRelativelyStaticSensorNetworks.InACMSASN,2003.

[17] S. Zhu, S. Setia, and S. Jajodia. LEAP: Efficient Security Mechanism for Large-Scale Distributed Sensor Networks. In ACM CCS, 2003.

[18] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An OnDemand Secure Routing Protocol Resilent to Byzantine Failures. InACM WiSe, 2002.

[19] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure OndemandRouting Protocol for Ad Hoc Networks. In ACM MOBICOM, 2002.

[20] Y.-C.Hu,D.B.Johnson, and A.Perrig.SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless AdHoc Networks .In IEEE Work-shop on Mobile Computing Systems and Applications (WMCSA'02),2002.
[21] P. Papadimitratos and Z. Haas. Secure Routing for Mobile AdHoc Networks. In Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS), 2002.
[22] C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. In IEEE SPNA, 2002.
[23] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical En-route Filtering of Injected False Data in Sensor Networks. In IEEE INFOCOM, 2004.
[24] S. Zhu, S. Setia, S. Jajodia, and P. Ning. An Interleaved Hop-by-Hop Authentication Scheme for Filtering False Data in Sensor Networks. InIEEE Symposium on Security and Privacy, 2004.
[25] N. S. U. Shankar and D. Wagner. Secure verification of location claims.In ACM WISE, 2003.
[26] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy. Secure loca- tions: Routing on trust and isolating compromised sensors in location- aware sensor networks. In ACM SENSYS, Poster Abstract, 2003.
[27] J.Hill,etal,"Systemarchitecturedirectionsfornetworkedsensors",inProceedingsofACMASPLOSIX,2000.
[28] [6] H. Chan, A. Perrig, "PIKE: Peer Intermediaries for Key Establishment in Sensor Networks", Proceedings of IEEE Infocom, 2005.