Routing Security in Sensor Network: HELLO Flood Attack and Defense

Md. Abdul Hamid, Md. Mamun-Or-Rashid and Choong Seon Hong*
Department of Computer Engineering, Kyung Hee University
Seocheon, Giheung, Yongin, Gyeongi 449-701 KOREA
hamid, mamun{@networking.khu.ac.kr}, cshong@khu.ac.kr

***Abstract*-We consider Wireless Sensor Network (WSN) security and focus our attention to tolerate damage caused by an adversary who has compromised deployed sensor node to modify, block, or inject packets. We adopt a probabilistic secret sharing protocol where secrets shared between two sensor nodes are not exposed to any other nodes. Adapting to WSN characteristics, we incorporate these secrets with bidirectional verification and multipath routing to multiple base stations to defend against HELLO flood attacks. We then analytically show that our defense mechanisms against HELLO flood attack can tolerate damage caused by an intruder.**

1. INTRODUCTION

In a large-scale sensor network individual sensors are subject to security compromise. Where the nature of communication is broadcast and, hence, an attacker can overhear messages posted by any sensor node; security is an important issue here. Wireless Sensor Networks (WSNs) are comprised of many small and resource constrained sensor nodes that are deployed in an environment to gather sensed data and forward that data to interested legal users.

Advances in micro-electro-mechanical systems (MEMS) technology allow sensors to be reprogrammable, self-localizing, and to support low-energy, wireless, multi-hop networking, while requiring only minimal pre-configuration. To support the reliability of coordinated control, management, and reporting functions, the sensor networks are self-organizing with both decentralized control and autonomous sensor behavior, resulting in a sophisticated processing capability [5].

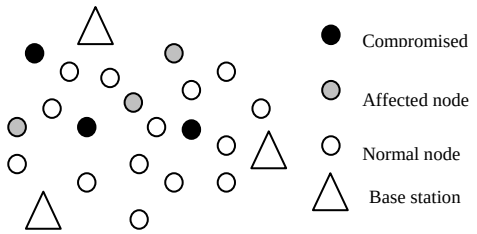


Fig. 1. In a sensor network, compromised nodes spoof, inject, modify, or represents false identity to affect normal sensor node to collect sensed data.

There are several network layer attacks against sensor networks and are well described in [3]. Among them, spoofed, altered, or replayed routing information, selective forwarding, sinkhole attacks, Sybil attacks, wormholes, HELLO flood attacks, acknowledgement spoofing are well known attacks that try to manipulate sensed data.

HELLO flood attack is introduced in [3]. Many protocols require nodes to broadcast HELLO packets to announce themselves to their neighbors, and a node receiving such a packet may assume that it is within (normal) radio range of the sender. This assumption may be false: a laptop-class attacker broadcasting routing or other information with large enough transmission power could convince every node in the network that the adversary is its neighbor. In this paper we consider routing security against HELLO flood attack.

2. CONTRIBUTION OF THE PAPER

The main contributions of the paper are as follows:

- We present probabilistic secret sharing protocol adopted from [1] where, a small increase in the number of secrets maintained by a user substantially reduces the probability of privacy compromise. And it is beneficial for the case where the sensor nodes do not have the capability to hold sufficient secret to ensure privacy. We show how these secrets can be used to route packets in a secured way.
- Then we propose defense mechanisms against HELLO flood attack using the secrets that nodes share among themselves.

3. ORGANIZATION OF THE PAPER

This paper is organized as follows. Section 4 describes related work. Section 5 discusses the network assumption and threat model and capabilities of sensor nodes. In section 6, the key assignment protocol is described in brief. Section 7 describes the defense against HELLO flood attack and addresses problem associated with this defense and section 8 addresses further defense to tolerate the damage. In section 9, we discuss about our counter measures against HELLO flood attacks and section 10 concludes the paper.

4. RELATED WORK

*Corresponding Author
This work was supported by ITRC Project of MIC

Sensor network security has been studied in recent years in a number of proposals. Kulkarni et al. [1] analyzes on the problem of assigning initial secrets to users in ad-hoc sensor networks to ensure authentication and privacy during their communication and points out possible ways of sharing the secrets.

Fan Ye et al. [2] focused on how to filter false data using collective secrets and thus preventing any single compromised node from breaking down the entire system. In [3] Karlof et al. thoroughly discussed the problem of secure data transmission for different routing protocols and they conclude that Many sensor network routing protocols have been proposed, but none of them have been designed with security as a goal. They suggested the security goals required for routing in sensor networks.

Passive attacks such as cipher text attack and chosen cipher text attacks, a security protocol has been proposed in [4] that ensures forward and backward secrecy of the session key, so that if any set of the session key is compromised, these compromised keys do not undermine neither the security of future session keys, nor the security of past session keys. Their works requires synchronization initiated by base station and also by sensor networks. SPINS [5] implements symmetric key cryptographic algorithms with delayed key disclosure on motes to establish secure communication channels between a base station and sensors within its range. Reference [6] uses public key algorithms, which are infeasible on small sensors of constrained computing, energy, and memory resources.

Our approach considers the routing vulnerability specially HELLO floods and discusses the counter measures and design considerations for secure routing in sensor networks.

5. Network Assumption And Threat Model

We consider a network composed of moderately large number of resource constrained sensor nodes. We further assume that the sensor nodes are deployed in high density, e.g. battlefield deployments. Each sensor node has a communication range such that if the distance between two sensors is more than this range, they can not communicate. We also assume that the communications channels are bidirectional, i.e. if a node *y* can receive a message from *z*, then it can also send a message to *z*.

We assume that an adversary can pose the following threat:

- An attacker can cause a HELLO flood attack by advertising a very high quality route to the base station. So, every node in the network could cause a large number of nodes to attempt to use this route thereby sending the legitimate packets beyond the actual destination.

6. Key Sharing Protocol

In this section, we present the probabilistic protocol, the tree protocol, for assigning the initial secrets. We will describe the single tree protocol and then compute the multiple trees based key assignment.

6.1 Secret instantiation by Tree Protocol

We present single tree and then multiple tree protocol. For each of these versions, we first identify the secret distribution protocol that determines the secrets that each user should get. Then, we present the secret selection protocol; when two users need to communicate, they use this protocol to determine a shared secret that they should use. Subsequently, we compute the probability of compromise.

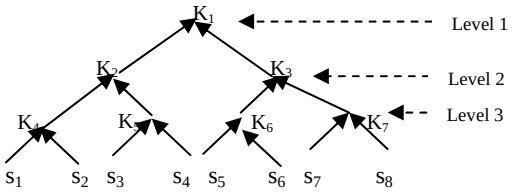


Fig. 2. Single tree key assignment

We organize the secrets in a tree (Fig. 2). Each non-leaf node is associated with a secret and each leaf is associated with a sensor node. Each sensor node is assigned an ID that identifies its location in the tree. Finally each sensor node is provided the secrets along the path towards the root. Thus, node s_1 has the secrets, k_1, k_2 and k_4.

When two nodes, say, s_1 and s_2, want to exchange messages during their effective communication, they first exchange their identities. Then, they identify their least common ancestor and based on the secret distribution mechanism, the common secret associated with this ancestor will be available to both s_1 and s_2. So, the secret associated with the ancestor will be used for communication between s_1 and s_2. For example, two nodes s_1 and s_2 want to communicate then they will use secret key k_4 whereas if s_1 and s_5 want to communicate then they will use secret key k_1.

6.2 Computing the vulnerability of security

Let *x* be an intruder that can observe the communication between any two arbitrary nodes *y* and *z*. We calculate what is the probability that *x* knows the shared secret that *y* and *z* use. During this analysis, let the degree of the secret-tree be *d*.

Now, we consider different cases based on the shared secret that *y* and *z* use during communication. First, we consider the probability that *y* and *z* use the secret at the root (level 1). Such a situation arises if *z* is not a descendant of the level-2-ancestor of *y*. Thus, the probability of this case is $d - 1/d$. And, in this case, the probability that *x* is aware of the secret that *y* and *z* use is 1; all users in the secret-tree have the secret associated with the root.

Next, we consider the probability that *y* and *z* use the secret at level 2 in the tree. Such a situation arises if *z* is a descendant of the level-2-ancestor of *y* and *z* is not a descendant of the level-3-ancestor of *y*. Thus, the probability

of this case is *1/d X (d - 1)/d*. Moreover, *x* is aware of the shared secret between *y* and *z* iff *x* is a descendant of the level-2-ancestor of *y*. Thus, the probability of this case is *1/d*.

Continuing thus, the probability of compromise, p_c, that *x* is aware of the shared secret used by *y* and *z* is:

$$p_c = \frac{d-1}{d} 1 \left(\sum_{i=0}^{l} (1/d)^{2i} \right)$$

$$= \frac{d-1}{d} 1 \left(\sum_{i=0}^{\infty} (1/d)^{2i} \right)$$

$$= \frac{d-1}{d} \frac{1}{d - 1/d^2}$$

$$= \frac{d}{d+1}$$

We see that as the degree of the tree, *d*, increases, the level of security decreases, also the number of secrets maintained by nodes decreases. So, to increase the level of security, we use multiple secret trees.

Multiple Tree Protocol: Secret distribution is the same as single tree protocol with one exception where the position of all sensor nodes is not identical. Because, Clearly, if we use two trees where the position of all users is identical and if *x* knows the secret (used by *y* and *z*) in the first tree then, by definition, *x* will know the secret in the second tree. Hence, when we use multiple trees to reduce the probability of compromise the probability that *x* knows the secret in one secret-tree should be independent of the probability that l knows the secret in another tree. This can be achieved if there is no correlation between the locations of a sensor node across two trees. Given *T* secret-trees, each with degree *d*, the probability that *x* knows secrets from all the trees is $(d/(d + 1))^T$.

We assume that a sensor node maintains *m* secrets in the multiple tree protocol. Thus, *m/log2n* trees are maintained. If we ignore the case of partial trees, the level of security with *m* secrets is $(2/3)^{(m/\log_2 n)}$, where the degree of the tree is 2.
So, we realize from the analysis, that the probability of compromise is of the form $g^{O(m)}$, $g<1$, *m* is the number of secrets that a sensor node maintains.

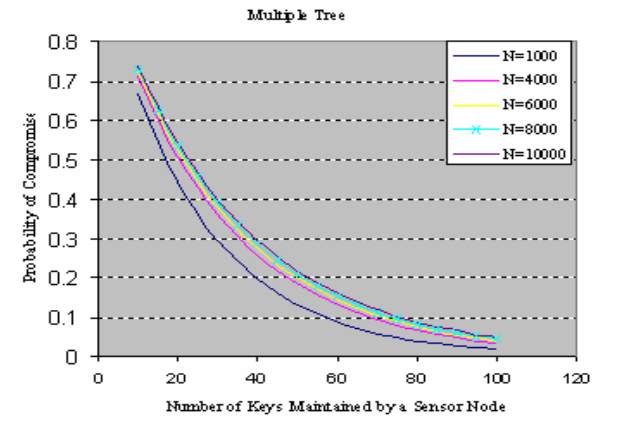


Fig. 3. Probability of security compromise vs. number of secret keys maintained by a sensor node. Each plot corresponds to the total number of sensor nodes N in the network.

Fig. 3 shows the effect of the number of secrets on the probability of security compromise in the multiple tree protocol. As can be seen, even if the number of sensor nodes is large, small number of secrets suffices to ensure that the probability of compromise is small.

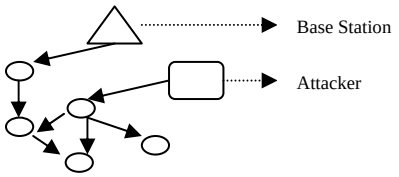
7. Counter Measure Against Hello Flood Attacks (Bidirectional Verification)

Many protocols require nodes to broadcast HELLO packets to announce themselves to their neighbors, and a node receiving such a packet may assume that it is within (normal) radio range of the sender. This assumption may be false: a laptop-class attacker broadcasting routing or other information with large enough transmission power could convince every node in the network that the adversary is its neighbor. To launch this kind of attack, an adversary's packet sending range must be bigger than a normal node's sending range.

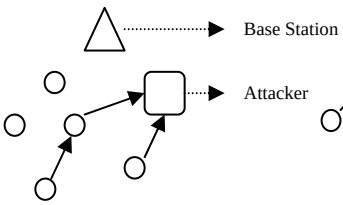
If each sensor node constructs a set of reachable neighbor nodes, and is only willing to receive REQ messages from this set of neighbor nodes, then REQ messages from an adversary transmitted with larger power will be ignored. Thus, the damage from a HELLO flood attack can be restricted within a small range.

To defend against attack, each request (REQ) message forwarded by a node is encrypted with a key. As we have shown from the tree protocol that any two sensor nodes share some common secrets, the new encryption key is generated on-the-fly (i.e. during communication). In this way, any node's reachable neighbors can decrypt and verify the REQ message while the attacker will not know the key and will be prevented from launching the attack. We show that the new key combined with the echo-back mechanism can well protect this attack.

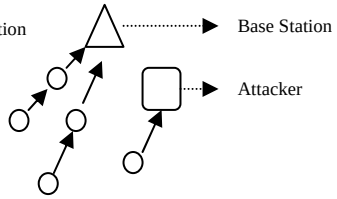
Fig. 4 gives a pictorial view of how HELLO flood attacks can be initiated and the defense against the attack. We see that the message exchange won't be blocked by an adversary when bidirectional verification is applied.



(a) Attacker advertises REQ message with large transmission power, legitimate nodes think that attacker is their initiator.



(b) HELLO flood attack, legitimate nodes think that attacker is their initiator



(c) With encrypted echo-back mechanism, message exchange won't be blocked by the attacker

Fig. 4. HELLO flood attack and defense

Each node locally broadcasts an echo message to its neighbor with format:

$$s_1 \rightarrow : \text{ECHO}||E_{\text{new-key}}(IDs_1||\text{nonce})$$

Where, ECHO is the message type, ID is the ID of the sensor node s_1, nonce is the random number. If a node, say, s_2 receives this message, it sends echo reply with format:

$$s_2 \rightarrow s_1 : \text{ECHOBACK}||E_{\text{new-key}}(IDs_2||\text{nonce}).$$

When node s_1 receives this message, it records node s_2 as its verified neighbor. If an attacker obtains the shared secrets after a node has received its new encrypted key, it can not know the new pairwise key. Computing the pairwise key is more robust and secure in multiple tree protocol as we have described earlier, where we have shown that the probability of compromise of a secret is very low. However, if an attacker obtains the new key, it can initiate echo-back many times by sending several echo messages. The attacker can generate false identities and can initiate Sybil attack, adding new nodes with false identities. To prevent such attacks, node should destroy its new key from memory after a certain time that is long enough to set up pairwise keys with all its neighbors. Again, during communication, it can calculate new key from the secrets they share.

7.1 Problem of Bidirectional verification

As we have stated that this defense against "HELLO flood" attack is to verify the bidirectionality of a link before taking meaningful action based on a message received over that link. But, this defense gets less effectiveness when an attacker has a highly sensitive receiver as well as a powerful transmitter. If an attacker compromises a node before the feedback message, it can block all its downstream nodes by simple dropping feed back messages. And thus, such an attacker can easily create a wormhole to every node within range of its transmitter/receiver. Since the links between these nodes and attacker are bidirectional, the above approach will unlikely be able to locally detect or prevent a "HELLO flood". We propose a different way of reliable exchange of messages among nodes and base stations. We show that when any particular node has different route to send data, this problem is solved.

8. Multi-path Multi-base Station Data Forwarding

We describe how a sensor node can forward its sensed data to multiple routes i.e. multiple base stations in case where an attacker manages to compromise a sensor node. We assume that, there are a number of base stations in the network who have control over specific number of nodes and also, there are common means of communications among base stations. Each base station has all the secrets those are shared by all the sensor nodes according to the key assignment protocol described earlier.

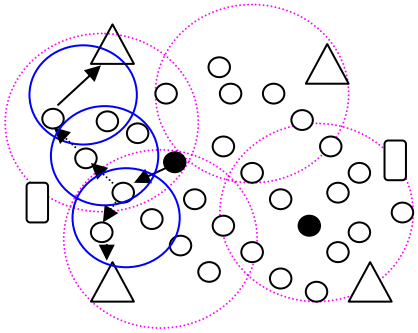


Fig. 5. Ordinary node gets REQ message from compromised node but does not forward message to it, rather it sends message to its verified neighbor by alternative routes.

Given the shared secrets and the generated new key between two sensor nodes, the operation of setting up different routing paths is as follows:

Step 1: As each sensor node shares some common keys according to the secret distribution protocol (i.e. Multiple Tree Protocol), every node uses the echo-back scheme to identify its neighbor nodes and sets up pairwise new key with its verified neighbor nodes. Then it uses its new key to exchange messages among them.

Step 2: Each base station broadcasts its request (REQ) message to its neighbor nodes with the following format:

$$\text{REQ}||ID_s||E_{key}(ID_B||\text{HCN})$$

Here, REQ is the message type, ID_s is the ID of the sending node s, ID_B is the base station ID who generated this request message, E_{key} is the key that is common between any node to which base station floods the message and HCN is the base

station's one-way hash chain number. Receiving node verifies that the REQ comes from the base station, then it forwards the REQ to its neighbor node, say, *y*, with the format:

$$REQ||ID_y||E_{new\text{-}key}(ID_B||HCN)$$

Step 3: When any ordinary node say, *y*, receives this REQ message, it checks the sender ID. If *s* is *y*'s verified neighbor, *y* decrypts and authenticates the sender with computed new key $E_{new\text{-}key}$. If the message sender is valid, it replaces the HCN with the new value and encrypts the REQ message with its $E_{new\text{-}key}$ and broadcasts the newly encrypted message.

As shown if figure 5, where four base stations with their communication range and sensor nodes with their communication range, if any message comes from a malicious node, the message won't be forwarded to that node, instead, the sensing node will take a different route to send data. Any base station, when receives the sensed data, it can cooperate with other base stations to interpret the sensed data as base station is powerful enough to communicate among themselves.

9. Discussion

In simple defense, we have shown every node to authenticate identity with shared secret by the means of bidirectional verification. We have shown that if the protocol sends the messages in both directions over the link between the nodes, HELLO floods are prevented.

We have shown a different approach when bidirectional verification does not prevent a compromised node. We present multi-path multi-base station routing. The flooding of REQ messages can securely establish direction without feedback to each base station. By setting up a new pairwise key from secret shared by nodes, multi-path routing improves intrusion tolerance. Specific one-way hash chain number (HCN) is addressed to defend against replay attack.

Placement of base stations depends on different applications with different constraints, e.g. number of total sensor nodes and number of base stations. Normally, base stations should be placed far away from each other to make the system resilient to node compromise.

Each sensor node needs to save shared keys, one-way hash chain number, and several random numbers. If each key is 64 bits long and a node communicates with *n* neighbors, keeps *r* random numbers, and there are *b* base stations, then the node needs *4X8X(2n + b +r + 2)* bytes to store all keys with 4 keys in each node. 992 bytes are needed if there are 4 base stations, 10 neighbor nodes, 5 random numbers. Current sensor nodes provide 4 KB SDRAM, 128 KB flash memory, 4KB embedded EEPROM, and 128K extended EEPROM. If the keys are not changed often, they can be stored in the 4KB embedded EEPROM.

Currently we are exploring the effect of node failure and the effectiveness of secure multipath setup. Maintenance issues like message loss, nodes joining and leaving are left as future works.

10. Conclusion

Our work described the defense against HELLO flood attack by introducing bidirectional verification and multi path routing using shared secret between sensor nodes. We have adopted a probabilistic key assignment among sensor nodes and during communication, each node can calculate a pairwise key using these common secrets and hence improving the network resilience against security threats. The key objective of our approach is to tolerate damage caused by an adversary who has captured deployed sensor nodes and is intent on injecting, modifying or blocking packets.

References

[1] S. S. Kulkarni, M. G. Gouda, and A. Arora, "Secret instantiation in ad-hoc networks," *Special Issue of Elsevier Journal of Computer Communications on Dependable Wireless Sensor Networks*, pp. 1–15, May 2005.

[2] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical En-Route Filtering of Injected False Data in Sensor Networks," IEEE Journal on Selected Areas in Communications, vol. 23, no. 4, April 2005.

[3] C. Karlof and D.Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, September 2003.

[4] R. Di Pietro, L. V. Mancini, and S. Jajodia, "Providing secrecy in key management protocols for large wireless sensors networks," *Journal of AdHoc Networks*, 1(4), pp.455-468, 2003.

[5] V. Wen, A. Perrig, and R. Szewczyk, "SPINS: Security suite for sensor networks," *in Proc. ACM MobiCom*, pp. 189–199, 2001.

[6] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and robust access control for mobile ad hoc networks," *Proc. IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 1049–1063, Oct. 2004.

[7] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, and et al., "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking," *Computer Networks (Elsevier), Special Issue on Military Communications Systems and Technologies*, 46(5):pp.605–634, December 2004.

[8] J.R. Douceur, The Sybil attack, in: 1st International Workshop on Peer-to-Peer Systems (IPTPS _02), 2002.

[9] Y. Hu, A. Perrig, and D. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," *In Second ACM Workshop on Wireless Security (WiSe'03)*, San Diego, CA, USA, September 2003.

[10] H. Chan, A. Perrig, D. Song, "Random key predistribution schemes for sensor networks," *IEEE Symposium on Security and Privacy*, 2003.

[11] W. Du, J. Deng, Y. Han, P. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," *ACM Conference on Computer and Communications Security (CCS)*, pp. 42–51, 2003.

[12] Y. Zhang, W. Lee, Intrusion detection in wireless ad hoc networks," *in: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pp. 275–283, 2000.

[13] S. Yi, P. Naldurg, R. Kravets, "Security-aware ad hocrouting for wireless networks," *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ACM Press, New York, pp. 299–302, 2001.

[14] D. Braginsky, D. Estrin, "Rumour routing algorithm for sensor networks," *in: First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[15] A. Manjeshwar, D. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," *in: 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, 2001.