Location Privacy and Authentication for Low-cost Sensor Node Devices Using Varying Identifiers

Md. Abdul Hamid and Prof. Choong Seon HONG
Department of Computer Engineering, Kyung Hee University
hamid@networking.khu.ac.kr, cshong@khu.ac.kr

[1]Abstract

Because a sensor node must operate on a tiny battery, the goal to eliminate energy inefficiencies leads the current researchers excavating for new techniques to advocate. As sensor networks edge closer towards wide spread deployment, security issues become a central concern. So far much research has focused on making sensor networks feasible and useful, and has not concentrated much on security issues especially computationally inexpensive techniques. In this paper we introduce a simple scheme relying on one-way hash-functions that greatly enhances location privacy by changing traceable identifiers on every read getting by with only a single, unreliable message exchange. Thereby the scheme is safe from many threats like eavesdropping, message interception, spoofing, and replay attacks.

1. Introduction

The wireless sensor network is receiving a lot of attention by the researcher due to recent advances in electronic and computer technologies. Sensor networks usually consist of a large number of ultra small autonomous devices, called a sensor node, is battery powered and equipped with integrated sensors, data processing capabilities and short-range radio communications. In typical application scenarios, sensor nodes are spread randomly over the terrain under scrutiny and collect sensor data. Sensor networks are being deployed for a wide variety of applications, including military sensing and target tracking, environment monitoring, patient monitoring and tracking, smart environments, scientific exploration, and monitoring of nuclear power plants etc. Security services such as authentication and key management are critical to secure the communication between sensors when sensor networks are deployed in a hostile environment, as they are prone to different types of malicious attacks. For example, an adversary can easily listen to the traffic, impersonate one of the network nodes, or intentionally provide misleading information to other nodes.

2. Preliminaries

We define g = h(x) as a cryptographic one-way hash function [1]. Ideally, besides the function being difficult to invert, the output g should not reveal any substantial information on its preimage x. In fact, this is often assumed in practice without mathematical justification [2]. Hence, use of "Secure Keyed One-Way Hash Functions" [3] or "K-hash Functions" [4] should be considered for maximum security. However, standard heuristic hash functions sufficiently hide information in practice [5]. The proposed scheme uses identifiers with limited validity as source for keys thus limiting security implications further. Since the number of gates in sensor nodes must be kept as small as possible for keeping the cost per piece low, an efficient implementation for the hash function in hardware is required [6]. Besides the hash function, a suitable conjunction function g = conj (a, b) is needed. It will be depicted with the "."-sign in the following (like g = a . b). We consider a simple exclusive-or function is adequate for the purpose.

Each sensor node needs to contain fields for the following entries:

- Current sensor node ID ("ID")
- Transaction number ("TN")
- Last successful transaction number ("LSN")
- Additional fields for user data or a master key

are conceivable if it is required.

The Database of the base station needs to contain a

[1] This work was supported by MIC

primary index of the table
- Current sensor node ID ("ID")
- Last transaction number ("TN")
- Last successful transaction number ("LSN")
- Associated DB entry ("AE")
- A reference to sensor data / user data ("DATA")

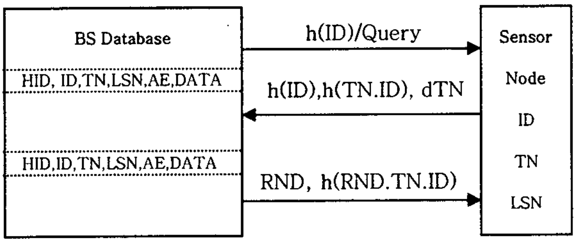

Figure 1: Message exchange

3. Initial setup

The data fields of the sensor node are initialized to the following values: The ID is set to a random value. The TN and the LSN are set to the same value which should be another random number. The corresponding row in the database of the base station must be created. The ID is set to the ID of the sensor node, the primary index HID is set to h(ID). The TN and LSN fields get the value of TN/LSN of the sensor node. The AE is not set since no associated entry exists initially.

4. Detailed description of the scheme

When the base station queries the sensor node, the node exposes no other information than the hash of its ID, namely h(ID), that is used for identifying and addressing the sensor node. When queried, the node increases its transaction number (TN) by one and sends the following information (Figure 1): h(ID), the hash h(TN.ID) and the difference between its current transaction number and the number of the last successful transaction (dTN=TN - TSN).

In this message, h(ID) identifies the sensor in the database of the base station with its stored H(ID). Here, h(TN.ID) has the purpose of counteracting replay attacks. It changes in every query and is checked by the legitimate receiver. Secondly, it authenticates the sensor node. Though conjugating the TN with the current ID is not mandatory, it may be useful if the number of bits used for TN is rather small.

dTN is used at the legitimate receiver to recalculate the current TN used by the sensor. Since it is only a table with the following entries for each record row:

- Hash of current sensor node ID ("HID"), acting as difference with a value '1' if no error has occurred, no information that could be utilized by an attacker for tracing the sensor node by its TN is revealed.

In the database, the record with HID=h(ID) is selected. The stored LSN and the received dTN are added together, thus obtaining the current TN of the sensor node (TN*) and the hash h(TN*.ID) is calculated. If the value does not match the one in the message (h(TN.ID)), the message is discarded. Still, if the message proves to be valid so far, the TN* and the stored TN are compared. If the TN* is not higher than the TN a replay attack is in progress and the message is discarded.

In case, if everything goes well, the TN* is stored as TN in the record row and the message is processed further. Now a random number RND is generated from the base station. With this number a new ID ('ID*') is created performing ID*=RND.ID. If an associated DB entry (AE) exists, the ID field of this record row is updated to the ID* and its HID is updated to the hash h(ID*). Otherwise, a new row is appended to the database inserting the ID* as ID and h(ID*) as HID and copying the reference data. The AE entry of the row is updated to the point to the other row and vice versa as well. The TN of the newly selected row is updated to the TN* value, its last successful transaction number (LSN) gets the same value.

Now a reply message containing RND and a hash h(RND.TN*.ID) is created and sent to the sensor node. The node checks the hash. If it is not correct the message is discarded and no further action is taken. Otherwise the sensor updates its stored ID to the value RND.ID and sets its last successful transaction number (LSN) to the TN value. Now the sensor node has a new ID whose hash h(ID) will be used as node identifier at the next query attempt.

5. Attacks

As the communication is performed over radio frequency, intercepting or blocking the request massage is a denial-of-service attack preventing sensor node identification. Loss, interception or blocking of the reply message results in preventing the node ID from being changed but fortunately has no other implications. The node will use its old ID in the next request which will match the unchanged table row in the base station database. A row in the database is never overwritten until the other entry has been addressed by the sensor providing that one

It is rather easy to detect error in message transfer afterwards on the basis of dTN value that is unequal to one. Suspiciously high values attract attention and counteractive measures can be approached. Similarly replay attacks can not compromise the scheme since the validity of messages is limited by means of the unique transaction numbers (TN). Any message of the sensor that reaches the database renders all previous messages invalid. Further, the sensor accepts only messages that are equipped with the current TN.

If the node's current ID or its previous ID in combination with the corresponding LSN value becomes known to an attacker he can imitate the node or wipe out the link between the node and the base station rendering the node unserviceable.

Eavesdropping is no issue as long as a malicious node (attacker) is not able to gain a current ID and TN value by means of cryptanalysis. Therefore, we stated before that he hash function should be selected in such a way that no usable information upon its preimage is revealed. Spoofing is not possible as the sensor node and the base station authenticate themselves by knowing the current ID and TN value.

6. Conclusion

The proposed scheme has a high inherent security rendering it a useful technique for many kinds of applications without relying on strong symmetric or even asymmetric encryption as those techniques are costly to implement and offers many more opportunities for attacks [7] because of stored long term secrets. Access control to data and changing other properties should be moved to the base station where plenty of computing power and the feasibility of certificate management is available inexpensively [8] Moreover, security systems and access control schemes can be changed easily according to the current requirements.

The main gain of the proposed scheme is its simplicity as it only requires a hash function in the sensor node and the data management at the base station. It offers high degree of location privacy and is resistant to many forms of attacks. Further, the communications channel need not be reliable and trusted.

Dynamic topology change and in case of new node joining/leaving the sensor networks are not considered here in our scheme and are left for the future works. being currently valid and the one to be overwritten being obsolete.

7. References

[1] Bakhtiari, S. et al.: "Cryptographic Hash Functions: A Survey," Technical Report 95-09, Department of Computer Science, University of Wollongong, 1995

[2] Canetti, R. et al.: "Perfectly One-Way Probabilistic Hash Functions," 30th Annual Symposium on Theory of Computing, pages 131-140, 1998

[3] Berson, T. et al.: "Secure, Keyed, and Collisionful Hash Functions," Technical Report SRI-CSL-94-08, SRI Int., 1994

[4] Bakhtiari, S. et al.: "Keyed Hash Functions, Cryptography: Policy and Algorithms," E. Dawson and Jovan Golic (Eds), Lecture Notes in Computer Science, vol. 1029, pp. 201-214, Springer, 1996

[5] Weis, S.: "Security and Privacy in Radio-Frequency Identification Devices", MIT, 2003

[6] Sarma, S. et al.: "Radio-Frequency Identification: Security Risks and Challenges," RSA Laboratories Cryptobytes, Vol. 6, No. 1, 2003

[7] Weingart, S.: "Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses," Cryptographic Hardware and Embedded Systems - CHES 2000, volume 1965, pages 302-317, Springer LNCS, 2000

[8] Agrawal. D. et al.: "Advances in Side-Channel Cryptanalysis," RSA Cryptobytes Volume 6, No. 1, 2003

[9] Dorothy Denning. "*Cryptography and Data Security*," Addison-Wesley, 1982.

[10] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J.D.Tygar. "Spins: Security protocols for sensor networks," *Wireless Networks*, 8:521 - 534, 2002.

[11] Philippe Bonnet, J. E. Gehrke, and Praveen Seshadri. "Towards sensor database systems," In *Second International Conference on Mobile Data Management*, 2001.