Security Analysis of Secure Data Aggregation Protocols in Wireless Sensor Networks

Triana Mugia Rahayu, Sang-Gon Lee*, Hoon-Jae Lee

Department of Ubiquitous IT, Division of Computer and Information Engineering
Dongseo University
Busan, Korea

gia.sutriadi@gmail.com, nok60@dongseo.ac.kr, hjlee@ dongseo.ac.kr

*Corresponding author

Abstract— **In order to conserve wireless sensor network (WSN) lifetime, data aggregation is applied. Some researchers consider the importance of security and propose secure data aggregation protocols. The essential of those secure approaches is to make sure that the aggregators aggregate the data in appropriate and secure way. In this paper we give the description of ESPDA (Energy-efficient and Secure Pattern-based Data Aggregation) and SRDA (Secure Reference-Based Data Aggregation) protocol that work on cluster-based WSN and the deep security analysis that are different from the previously presented one.**

Keywords— **Data aggregation protocol, secure data aggregation protocol, ESPDA, SRDA, WSN**

I. Introduction

The wide range applications of WSN from military to civilian applications have proved that this type of network is significantly important. This network can contains hundreds, even thousands, of small-size sensor devices that are deployed in remote or hostile area. These sensors are resource-constrained devices which are limited in computation, power and memory.

One major challenge in WSN is how to preserve network lifetime. In order to reduce network energy consumption, network is divided into clusters. In each cluster, there is one cluster head (CH) that has responsibility to relay sensor readings of the cluster members to the base station (BS). By doing so not all sensor nodes are required to send data directly to BS thus network energy consumption is reduced. Eliminating the redundancy in the sensed data can also save the amount of energy needed for transmission. Data aggregation helps reducing the redundant data and even further saving network resources.

But then the need to secure the data aggregation schemes rises up because an aggregated data is actually the summary of the readings of some part of the network. Some researchers who consider the importance of the security aspect proposed secure data aggregation protocols for WSN to meet the security requirements of this network.

In the next section the security requirements of WSN are presented along with their relation to secure data aggregation scheme. Section III gives the details of two secure data aggregation protocols which are ESPDA [1] and SRDA [2]. Section IV gives the deep security analysis of those protocols. And section V concludes our findings.

II. Security Requirements of WSN

The security requirements of WSN are similar to those of traditional wireless network since they share some properties. But due to hostile environment and resource-constrained sensors, it is more challenging to devise protocols satisfying these requirements for WSN. Ozdemir *et. al.* [3] explained the required security properties of WSN and their interaction with data aggregation process.

- Data confidentiality – To ensure that the content of the message should not be revealed to the unauthorized receiver. Some secure data aggregation schemes provide this property in hop-by-hop basis in which any aggregator node needs to decrypt the received encrypted data before applying the aggregate function on it and then encrypt the aggregate data before transmitting it to the higher level aggregator or directly to the base station. While the other schemes provide end-to-end data confidentiality in which any aggregator node directly apply the aggregation function to the received encrypted data.
- Data integrity and freshness – Data integrity guarantees that the message has not been altered during the propagation. But if data aggregation is employed then it is not possible to have end-to-end data integrity since data aggregation yields in alteration. Data freshness protects data aggregation from reply attack.
- Source authentication – Enables sensor node to ensure the identity of the peer node that it is communicating with. A compromised node can launch Sybil attack in which it may send data under several fake identities in order to corrupt the aggregated data.
- Availability – To guarantee the survivability of network services against Denial-of-Service attacks. The attack aiming at an aggregator can make some part of the network losses its availability because the aggregator is responsible to provide the measurement of that network part.

III. PROTOCOLS DESCRIPTION

Secure data aggregation protocols described here are ESPDA and SRDA.

A. ESPDA

The preliminary versions of this protocol appear in [4, 5]. The main idea of ESPDA is that instead of directly sending the actual sensed data to the CH, the sensor nodes send their pattern codes at first. The pattern code is a representative data for the sensed data and is generated based on the secret pattern seed which is periodically distributed by the CH. The CH then compares those pattern codes, selects only the unique ones and requests the actual data from the nodes having the corresponding unique codes. In reply, each selected node then sends the encrypted actual data to the CH while the rest may be noticed to drop the data. The CH needs not to decrypt the received encrypted data because the data aggregation process is done prior to the actual data transmission. This reduces the overhead of the CH and thus contributes to the energy efficiency. Then CH can forward the message to BS.

Each sensor node is assigned a unique ID (id_i), a node specific secret key (k_i) and a secret key common to all nodes (k) prior to the deployment by BS. In addition, BS periodically broadcasts a random session number (r_b) in encrypted format using key k ($enc_k(r_b)$). Upon receiving r_b any node i computes the node specific session key ($k_{i,b}$) for data communication by XOR-ing its built-in secret key k_i with r_b, $k_{i,b} = k_i \otimes r_b$. Later on node i uses $k_{i,b}$ to encrypt its actual data. Accompanying this encrypted data, node i also sends its timestamp and ID number. Those two data will help BS to choose the right r_b and compute the right $k_{i,b}$ to decrypt the message. In order to provide data integrity, message authentication code (MAC) of the message using $k_{i,b}$ is also included in the message.

This protocol uses Nonblocking OVSF Block-Hopping (NOVSF-BH) technique. In which this technique improves the security and the spectral efficiency of network. In NOVSF codes, each OVSF code has 64 time slots such that any number of this timeslots can be assigned to a channel. The proposed NOVSF-BH technique assigns data blocks to time slots using different mapping in every session. So besides equipping every sensor node with keys, BS also periodically sends a different mapping permutation in encrypted format using key k ($enc_k(permutation)$) to CHs. This mapping permutation allows every node to map its data blocks according to the given code before sending it to the CH. By doing so the intruder first has to find the mapping pattern for that particular session and then try to decrypt the message.

Figure 1 shows the summary of ESPDA protocol. This figure depicts the protocol run in one session. In every new session BS broadcasts a new r_b. It encrypts r_b using key k. BS also sends to CHs the mapping permutations in encrypted form using key k. On how this mapping permutation is then distributed by each CH to its cluster was not explained in [1]. As with the possession of network key k, this task can be trivial. Each CH may broadcast it to its own cluster securely in encrypted form and with MAC computed using key k to guarantee the confidentiality, authenticity and integrity. Also each CH broadcasts to its cluster the encrypted pattern seed computed using key k.

1. i : compute $k_{i,b} = k_i \otimes r_b$
2. $i \rightarrow H$: $pattern\_code_i, timestamp_i, id_i$
3. H : compare pattern codes and select unique pattern codes based on timestamps
4. $H \rightarrow i_{selected}$: actual data request message
5. $H \Rightarrow i_{de-selected}$: ack message to discard the data (optional)
6. $i_{selected} \rightarrow H$: $id_i, timestamp_i, enc_{k_{i,b}}(d_i), mac_{k_{i,b}}(d_i)$
7. $H \rightarrow BS$: $id_H, id_i, timestamp_i, enc_{k_{i,b}}(d_i), mac_{k_{i,b}}(d_i)$

Figure 1. ESPDA protocol

Various symbols denote:

i, H, BS	:	A sensor node i, a CH and BS, respectively
$\Rightarrow, \rightarrow$	:	Broadcast and unicast transmissions, respectively
id_i	:	The id of node i
d_i	:	Sensed data from node i
$pattern\_code_i, timestamp_i$	:	The pattern code and timestamp of node i
$k_i, r_b, k_{i,b}$	:	the secret built-in key of node i, the random session number from BS and node specific session key of node i, respectively
$mac_k()$	:	MAC calculated using k key
$i_{selected}$, $i_{de-selected}$	:	The selected set of node that has unique pattern codes and unselected ones, respectively

B. SRDA

Similar to ESPDA, SRDA also consider data communication security protocol to work with data aggregation. The idea is that the actual sensor data is at first compared with the reference value and SRDA transmits only the differential value to the CH in encrypted form. The differential aggregation has great potential to reduce the amount of data to be transmitted from sensor nodes to CH.

SRDA uses random key predistribution protocol that is based on Eschenauer and Gligor's work [6] and takes advantage of the estimated location information of sensors which can be predicted with some probability from the way sensors are deployed. In short, this key distribution process makes every node in the network to share common keys with other nodes that are not physically located very far from it.

Figure 2 presents SRDA protocol after the key distribution process is implemented. In every new session, any node i computes its reference value M_1 by taking the average of last N sensed data value, where $N \geq 1$. The node i then sends encrypted M_1 to CH. It uses a secret key shared with CH, k_r. CH creates a reference entry for the node i with value M_1. For the subsequent transmission, node i then transmits only the encrypted differential value of the next raw data to the reference value, $M_j - M_1$ where $j \geq 2$. If a new session is started, CH removes the correspondent reference entry. The same concept is applied to CH when it sends the data to higher CH or BS.

```
1. $i$            : Computes reference value for the new session, $M_1$
2. $i \rightarrow H$  : $enc_{k_r}(M_1)$
3. $H$            : Create reference entry for node $i$ with value $M_1$
4. $i \rightarrow H$  : $enc_{k_r}(M_j - M_1)$, where $j \geq 2$
5. $H$            : if new session starts, remove correspondent reference entry
```

Figure 2. SRDA protocol

SRDA also considers to gradually increasing the security level of data packet as it travels to the higher level of clustering hierarchy. Data packet at higher level may contain summary of a large number of transmission from lower levels. SRDA uses RC6 because it is a parameterized algorithm where the block size, the key size and the number of rounds are variable. The security strength of the RC6 can be measured by *Security Margin*. The security margin of an encryption algorithm is the percentage of the deviation of the actual number of the encryption rounds from the minimum number of rounds for which the algorithm is considered to be secure. In order to save energy SRDA uses smaller security margin for lower level CHs compared to those at higher level.

IV. Security Analysis

In [3] authors summarized the security properties provided by several secure data aggregation protocols. ESPDA and SRDA provide data confidentiality, data integrity and source authentication. Their drawbacks are also presented. The ESPDA and SRDA do not allow intermediate nodes to perform data aggregation thus limits the benefit of data aggregation.

In this paper we present our security analysis on each secure aggregation protocol described in section III, highlighting the pitfalls of each protocol that are different from those presented in [3].

A. ESPDA

The drawbacks of EPDA protocol are explained in the following points:

- The use of static network key k is risky. If any intruder is able to successfully obtain the knowledge of key k then security of the protocol can be destroyed. Therefore rekeying for network key k is needed to mitigate this problem.
- The network key k is statically used over ever changing sessions. This key is used to encrypt the r_b, pattern seed and mapping permutations. Because of that the freshness of this information is not provided by ESPDA. Though it is not clear how the source authentication and message integrity are provided, but let us assume that the ID of the sender is included to the message as well as the MAC calculated using key k. The sender's ID and MAC help the receiver recognize the origin of message and verify the message integrity. But this protocol is still vulnerable to replay attack. Any intruder may rebroadcast the previously captured messages for distributing r_b, pattern seed and mapping permutations in another different session.
  The impacts are the nodes may have different $k_{i,b}$ with that of BS which may fail the decryption of the message by BS, some nodes may have different pattern seed with the rest of the cluster which result in not accurate data aggregation, and CH is not able to reconstruct the proper order of data blocks because of the different mapping permutation used by the member nodes. Dynamic session dependent group key can be a possible solution for this drawback.
- The data integrity of pattern code message from sensor node to CH is not provided. Any intruder can alter the pattern code message and disrupt the data aggregation. The integrity of the message from sensor nodes cannot be verified at CH but only at BS. Thus early detection cannot be afforded and this may not be energy efficient. Network key k can be used to calculate MAC of the message to provide data integrity.
- The source authentication only exists between node and BS, not between node and CH. Any intruder may launch impersonation attack in which any unauthenticated node may impersonate any legal node. Though the intruder does not have the legal shared secret key with BS, it can send its pattern code to CH. CH runs the pattern comparison algorithm with the received pattern code and requests the actual data to that illegal node if its pattern code is unique. Because an illegal data insertion cannot be detected by CH and reaches BS, it consumes network resources. Because CH cannot detect that earlier. Providing shared secret among nodes may become a possible solution.
- In the protocol, CH requests the selected nodes for their actual data in multi-unicast communication. This can be expensive. One broadcast message using broadcast authentication may be a solution.

B. ESPDA

The downside of SRDA is that BS still trusts the CH. There is no way for the BS to verify that the aggregated data come from sensors' readings. Any malicious CH can send some fake aggregated data without being detected.

V. Conclusions

In this paper we describe our security analysis of two secure data aggregation protocols, ESPDA and SRDA. Our security analysis presented here is different and deeper than the ones mentioned in previous work before. Some possible directions are also proposed besides highlighting the drawbacks of the protocols.

ACKNOWLEDGMENT

This is the statement of acknowledgement for ICACT2014 only for the third author.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology. (Grant number: 2013-071188). And it also was supported by the BB21 project of Busan Metropolitan City.

REFERENCES

[1] H. Cam, S. Ozdemir, P. Nair, D. Muthuavinashiappan, H. O. Sanli, Energy-efficient and secure pattern based data aggregation for wireless sensor networks, Comput. Comm., Elsevier 29 (4) (2006) 446-455.

[2] H. O. Sanli, S. Ozdemir, H. Cam. SRDA: secure reference-based data aggregation protocol for wireless sensor networks, in: Proceedings of the IEEE VTC Fall Conference, Los Angeles, CA, 26-29 September 2004, pp. 4650-4654.

[3] S. Ozdemir, Y. Xiao. Secure data aggregation in wireless sensor networks: A comprehensive overview, Comput. Networks, Elsevier 53 (2009) 2022-2037.

[4] H. Cam, S. Ozdemir, D. Muthuavinashiappan, and Prashant Nair, Energy-Efficient security protocol for Wireless Sensor Networks, IEEE VTC Fall 2003 Conf., October 2003, pp. 2981-2984.

[5] H. Cam, S. Ozdemir, Prashant Nair and D. Muthuavinashiappan, ESPDA: energy-efficient and secure pattern-based data aggregation for wireless sensor networks, IEEE Sensors- The Second IEEE Conf. on Sensors, Oct. 2003, pp. 732-736.

[6] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks”, *Proceedingsof the 9th ACM conference on Computer and communications security,* Washington, DC, USA, November 18-22 2002, pp. 41-47.