

Securing Data Aggregation against False Data Injection in Wireless Sensor Networks

Anuparp Boonsongsrikul, Kyung-suk Lhee and ManPyo Hong
Digital Vaccine and Internet Immune System Lab
Graduate School of Information and Communication
Ajou University, South Korea
E-mail: {anuparp, klhee, mphone}@ajou.ac.kr

Abstract—Data aggregation is an efficient technique for reducing communication cost in wireless sensor networks. However it is vulnerable to false data injection attacks where the adversary node sends false data to the aggregation node and thus, stops the base station to acquire correct data. This paper proposes a secured data aggregation protocol which detects the false data injection during the data aggregation and identifies the adversary node. The protocol is lightweight because it requires only a minimal cryptographic techniques, and takes $O(n)$ time for both data aggregation and tracking back the attacker.

Index Terms—sensor networks, data aggregation, security, false data injection attack, tracking back the attacker

I. INTRODUCTION

Data aggregation is widely used in wireless sensor network to reduce communication cost, but it also introduces many security problems. One of the problems is that a compromised node may inject false data in order to adversely influence in computing the final aggregation value. In addition, it is difficult to find which node injected such data because the data source becomes unknown during the in-network processing.

Many solutions were proposed to detect the false data injection during data aggregation. However, most of them require complicated communication protocols or cryptographic techniques, thereby adding more computation and communication burden to sensor nodes. Those solutions might be too expensive for some sensors with limited resources (slow CPU, small memory, weak transceiver, and small battery capacity), substantially reducing the sensor network's application functionalities.

In addition, most solutions can only detect the occurrence of an attack but cannot find the attacking node. Therefore an attacker may repeat the attack until he succeeds, each time adjusting the injected data. In this paper we propose a solution to detect the false data injection during the data aggregation and to find the attack node. Our objective is to devise a lightweight solution so as to be applicable to most sensors.

While most existing solutions achieve message authentication, integrity, or confidentiality by cryptographic means, we try to achieve similar goals by exploiting the two common characteristics of wireless sensor network. One is that sensors communicate wirelessly, so each node can hear (snoop) his neighbor's communication. The other is that most sensors are cheap, so we can afford hardware redundancy.

Our solution is lightweight because it requires only a minimal cryptographic techniques, and takes $O(n)$ time for both data aggregation and tracing back an attacker.

Our main idea is monitoring by children and judgment by majority. We propose the message formats of data aggregation against false data injection in wireless sensor networks. Let each sensor node reports anomaly to the base station, and let the base station retrieve the truth in reports. The base station can afford complicated, expensive analysis algorithm. Our objective is to provide lightweight technique for sensor nodes to report.

The base station can authenticate an attestation message with the attached MAC, but the base station is not given any other evidence to trust the message content. Instead, the base station detects an attack if he receive sufficient number of attestation messages pointing at the attacking node (*decision by majority*). An attacker can neither successfully inject a false data during the aggregation process nor frame an innocent node by sending fake attestation message, unless he compromises sufficient number of sensor nodes.

Even, monitoring by parent is feasible but difficult to attest because the parent can be the adversary. Moreover, peer monitoring is difficult to implement because sensor nodes do not know about each other. However, we consider that security of external nodes is unimportant in that they do not contribute to the data aggregation.

II. RELATED WORK

A significant amount of research effort has been focused on secure data aggregation techniques. However, to maintain the focus, we describe only those techniques that are related to false data injection attacks during data aggregation.

Przydatek [1] proposed a secure information aggregation (SIA) which provides a statistical security property under the assumption of a single aggregator. The goal is to prevent stealthy attacks. If a reported aggregation result is *close* to a true aggregation value with high probability then the home server will accept the reported aggregation result, otherwise the home server will reject it.

Wagner [2] proposed Resilient Aggregation in sensor networks. The base station uses aggregation functions to summarize the majority of readings. The median is a good summary statistic. For example, if an extreme value injected by the

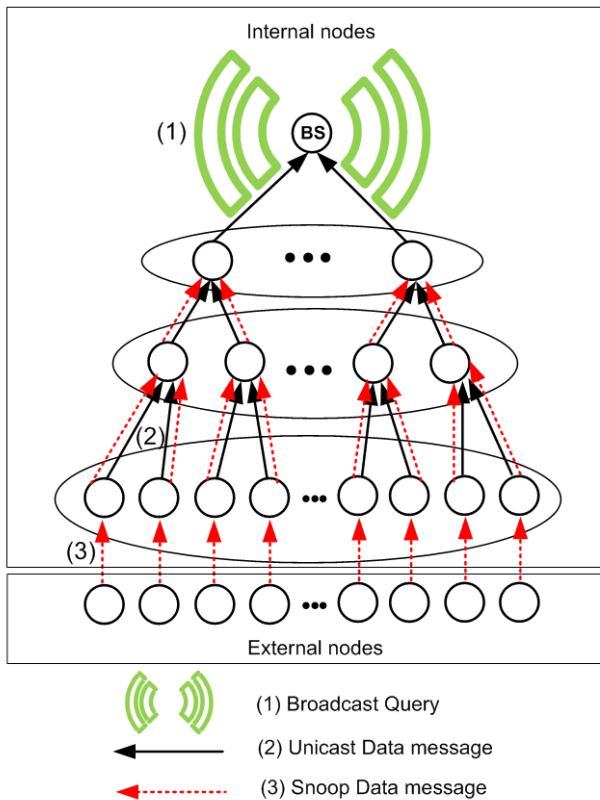


Fig. 1. Network Model

attacker is occurred, then such an extreme will be discarded because of the median function. However, median may not be suitable for some applications, for example multi-sensor fire alarm systems.

Yang [3] proposed a secure hop-by-hop data aggregation protocol (SDAP). SDAP uses a probabilistic grouping technique to divide sensor nodes into multiple logical groups. In each group, data aggregation does the same as TAG [4]. For detecting the attackers, when the base station receives an aggregate which is an outlier from others. The base station will ask to the group which provides an outlier. If the aggregated result between before and after asking is inconsistency, this means that there is the attacker. To improve the detection, it might need multiple paths for attestation. However, this leads the communication overhead to detect the attacker.

Chan [5] proposed securing Hierarchical In-Network Aggregation in Sensor Networks against stealthy attacks as same as [1], [3]. In data aggregation, sensor nodes iteratively construct a commitment structure resembling a hash tree. The base station will accept the aggregation result if all the sensor nodes have confirmed that aggregation structure is consistent with their data values; otherwise the base station will reject the aggregation result.

Vu [6] proposed THreshold security for Information aggregation in Sensor networks (THIS). For data aggregation, each sensor node compare its reading with an aggregated value produced from the aggregator. If close enough, the sensor

node replies to the aggregator with its signature; otherwise the sensor node ignores replying. The aggregator collects signatures from sensor nodes, and sends the aggregation value and all signatures to the base station. The base station receives the message containing the aggregation value and signatures. If the number of signatures is less than the threshold, the base station discards this message.

He [7] proposed PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks. The algorithms defense against the attacker who performs eavesdropping data. Each node hides its private data by slicing it into pieces. It sends encrypted data slices to different intermediate aggregators. After the pieces are received, intermediate nodes calculate intermediate aggregate values and further aggregate them to the base station.

Mlaih [8] proposed Secure Hop-by-Hop Aggregation of End-to-End Concealed Data in Wireless Sensor Networks. The authors proposed a pair of aggregated values; one is for authentication of hop-by-hop and the other one is for authentication of end-to-end. Besides such a pair, the ID list representing sensor nodes involves in a pair of aggregated values is produced and used to regenerate MAC to check integrity along the way at the base station. This paper is promising to get caught the attacker as same as our work. However, the concatenation of IDs will occur the overhead of data aggregation ,particularly in scalability.

III. ASSUMPTIONS

Each node shares a unique pairwise key with the base station for authenticated communication between them. Such keys are pre-installed to nodes before the sensor network is deployed.

The proposed solution uses a simple aggregation tree such as a spanning tree constructed by breadth-first search as in EN-CAST [9]. The sink node floods a discovery message, which is propagated throughout the network. After the flooding, every node knows who are its immediate neighbor nodes and can form a shortest path tree by selecting one of them as its parent.

Aggregation tree may be reconstructed periodically, and the base station does not know the topology of the tree (it would be too costly for the base station to learn such information).

The proposed solution requires two kinds of pairwise keys. One is shared between a node and the base station so that the base station can authenticate his message. Such keys are installed before the network is deployed, hence do not require any run-time establishment or management. The other is shared between a node and his parent so that the parent can authenticate his message. Although these keys do require run-time establishment and management, most existing solutions build more expensive cryptosystem on top of them.

In addition, we make more assumptions as follows.

1) *Unique node id*: Each node has only one id. For example, if a node is known as x to his children then he must be known as x to his parent as well.

This assumption might seem unnecessary, but a node might have multiple ids if the tree algorithm naively allows a (compromised) node to advertise multiple id's during the tree

construction. The need for this assumption will be explained in section V-B.

2) *External nodes*: During the aggregation, external (leaf) nodes behave differently from internal nodes: they do not contribute to the aggregation (i.e., they do not send their sensor reading values to their parents). Their only role is to monitor their parent nodes.

The reason for this assumption is to force all the nodes who contribute in the data aggregation to become internal nodes. Such external nodes make our detection strategy, *monitoring by children*, easier to implement. We will explain more about this in section IV-C.

A downside of this assumption is the hardware redundancy. For example, a full d-ary aggregation tree would need about twice the number of sensor nodes than usual.

However, our intension is to devise a simpler algorithm and more lightweight protocol at the expense of hardware redundancy. Considering that most sensor hardware is resource-constraint (slow CPU, small memory, limited bandwidth) but very cheap, such trade-off may often be beneficial.

3) *Node density*: There are sufficient number of children who are close to their parent and hence snoop his message.

4) *Parent-children authentication*: During an aggregation tree construction, each node establish a pairwise key with his parent node for authenticated communication between them. For this we may use an existing protocol such as Needham-Shroeder style key exchange or PIKE [10].

IV. PROPOSED PROTOCOL

The propose protocol uses the network model illustrated in Figure 1. It consists of the following three phases: 1) the query dissemination phase, where the base station initiate the aggregation, 2) the aggregation phase, where all nodes perform the aggregation, and 3) the attestation phase, where suspecting nodes send verification messages to the base station to find suspicious nodes and verify them. The message formats are described in table I and II.

A. Query dissemination phase

The base station initiates a data aggregation by broadcasting a aggregation query message to all nodes.

We do not provide an authenticated broadcast mechanism but instead we may use an existing, lightweight technique such as μ TESLA [11]. Note that an attacker can only drain the batteries of sensors with fake aggregation query messages, which is a different kind of attack (it is effectively a denial-of service attack) and therefore is out of the scope of this paper.

B. Aggregation phase

Every internal node sends an aggregation message to his parent in the aggregation tree. The data value in the aggregation message is computed by aggregating his own reading value and the data values of his children.

External nodes do read their data values but do not send aggregation message. Their only role is to monitor their parent nodes, and will be involved in the attestation phase.

Type of message	Format
query	<sid, type>
aggregate	<op, id, pid, data, AMAC>
attest	<op, id, pid, data, pdata, BMAC>

TABLE I
THE MESSAGE FORMATS.

Field	Meaning
type	1-bit value specifying the types of aggregation functions (sum or mean)
op	1-bit value specifying the message type (aggregate or attest)
sid	unique session identifier
id	node identifier
pid	identifier of parent node
data	aggregated value of the node
pdata	aggregated value of the parent which child nodes snoop
AMAC	AMAC($id sid, KEY_{id,pid}$) authenticates data message to parent node. It is computed using the concatenation of id and sid (sid is used as a salt value) and the pairwise key shared between the node and his parent. KEY is a pairwise key shared between the node and his parent.
BMAC	BMAC($id pid data sid, KEY_{id,BS}$) authenticates attestation message to the base station. It is computed using the id, pid, and data so that the sender cannot lie about these values. Key is a pairwise key shared between the node and the base station.

TABLE II
THE FIELDS OF MESSAGES.

C. Attestation phase

Parent node receives data from the child nodes. It aggregates them and forwards the aggregated message to the node at upper level in the tree. While forwarding, the child node overhears the forwarding value and compares it to its reported value. If the difference between them lies under a certain range, then the forwarded value is considered as normal. Otherwise, the parent node is considered as suspicious and thus, the child node reports about the suspicious parent node to the base station. The message that is used by child node is referred as attestation message.

In general, every child node that monitors the adversary node sends the attestation message to the base station. If the base station receives sufficient number of attestation messages which is not less than the number of node degree of the tree, then the base station uses the attestation messages to build a tree which is referred as attestation tree. The attestation tree is useful to locate the adversary (Figure. 2 is an example of attestation tree). The base station analyzes the attestation tree in the following aspects.

1) *The adversary may send information inconsistent to its child nodes*: For instance, in Figure. 2, if the base station derives a partial data <id, pid, data, pdata> from the attestation message of node 7-10 : <7, 3, 45, 45>, <8, 7, 45, 58>, <9, 7, 45, 58> and <10, 7, 45, 58> respectively, then the base

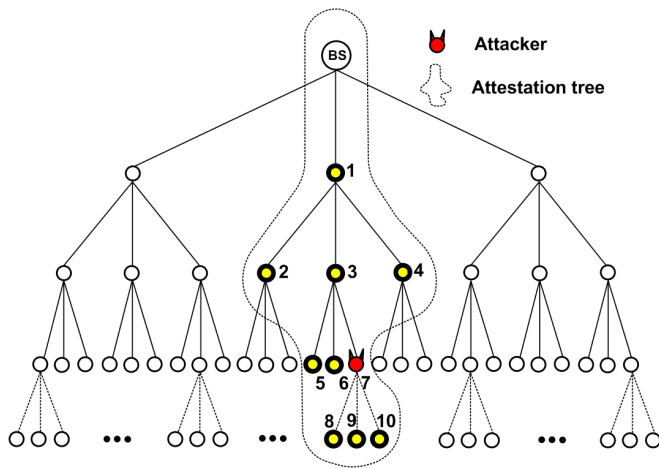


Fig. 2. The attestation tree with node degree = 3

station can detect that node 7 is the liar because node 8-10 report that data value of node 7 is 58 while node 7 reports that its data value is 45.

2) *The most suspicious node is the parent of leaf nodes of the attestation tree* : Let us call this suspicious node as node A. To double-check that node A is the attacker, the base station will compare data values between node A and its children who report the attestation messages. If the data value of node A is the outlier, then node A is the attacker.

For instance, Chauvenet's criterion [12] can be applied to detect outliers. Consider a set of data values is $\{46, 48, 44, 38, 45, 47, 58, 44, 45, 43\}$. The mean, $\bar{x}$ is 45.8 and the standard deviation, σ_x is 5.1. Note that the value 58 seems anomalously large and differs from $\bar{x}$ 45.8 by 12.2 or $2.4\sigma_x$. The probability that a value differs from $\bar{x}$ by $2.4\sigma_x$ or more is 0.016. In ten data values, we would expect to find only 0.16 (0.016×10) of one value worse than our suspect result. Since $0.16 < 0.5$, according to Chauvenet's criterion, the data value of 58 is the outlier. The node which provides the outlier is the attacker.

V. SECURITY ANALYSIS

A. Finding the attack node

If an attacker is indeed present, then he must be the parent of leaf nodes of the attestation tree. Therefore the base station gathers aggregation messages from these nodes, compares the data values in those messages, and finds the outlier using algorithms such as [13], [12], or Grubbs' test [14] etc.

As illustrating in Figure 2, if the abnormality of a data value is slightly high, then the grandparent node 3 and his siblings node 5 and 6 will also be affected when calculating aggregated values and therefore send attestation messages. In the worst case, all the nodes on the path from children and siblings of the abnormal node to the base station, node 1 - 10, will send attestation messages.

If there is only one attacker, then he must be the parent of leaf nodes of the attestation tree because children of the

attacker will send attestation messages. External nodes (leaf nodes in the aggregation tree) cannot be the attacker because they do not contribute to the aggregation. Therefore the base station only needs to compare the data values of the leaf nodes (in the attestation tree) and that of their parent to find the outlier.

Although it is possible that more attackers may reside elsewhere in the path, it does not pose a serious problem in the sense that our protocol detects (and hence can eliminate) the attacker. Other attackers will be detected in the subsequent aggregation sessions.

It is also possible that there are multiple paths if there are many attackers. The base station can simply choose a path and detect (eliminate) an attacker one at a time.

B. Uniqueness of ID

The snooping is not possible if a node can have multiple id's. For example, if a node is known as x to his children but known as y to his parent, then such a node cannot be monitored by his children. Such an anomaly would be possible if the aggregation tree algorithm naively allows a (compromised) node to advertise multiple id's during the tree construction. Therefore we assume that every node is guaranteed to be assigned a single id during the aggregation tree construction.

Note that a child cannot verify that the parent sends the aggregation aggregated message to the grandparent. Snooping is ineffective on this because the parent can send valid data value but with invalid MAC to fail the authentication. In other words, children cannot know if the parent makes progress or not.

C. Reporting false information

An attacker might include abnormal data value in the aggregation message but include normal data value in the attestation message so as not to get caught. Note that, however, the attestation message contains the data value of his parent as well as the data value of his own. If the attacker lies as such, many attestation messages of his children will testify him as a liar (*monitoring by children and judgment by majority*).

Also, an attacker might include false pid and/or abnormal value in $pdata$ to frame an innocent node. However, the base station will not regard a certain node as an attacker unless sufficient number of messages are pointing at him (*decision by majority*). Note that attestation message is authenticated using BMAC in order to prevent Sybil [15] attack by one or few compromised nodes.

D. Not sending attestation message

An attacker might choose not to send the attestation message because he cannot report false information during the attestation phase. However, the base station will still learn of his information because his children include his information in their attestation messages.

The base station cannot detect if an external node such as one of node 8-10 in Figure 2 does not send an attestation message. This however does not pose a problem, because

external nodes do not contribute to the aggregation hence cannot inject false data. Monitoring the parent of the missing external node is not a problem either, unless majority of the children of the parent are compromised.

VI. COMMUNICATION OVERHEAD

The communication overhead is measured as the number of hops all the messages traveled during an aggregation session.

A query dissemination message can be flooded to all nodes with $O(n)$ number of hops.

The aggregation phase requires $O(n)$ number of hops because each node sends one aggregation message.

The number of messages generated in the attestation phase depends on the number of attackers and their location. In this section we consider a simple case where there is only one attacker.

As we mentioned in section IV-C, if the abnormality of a data value is high enough then all the nodes (and their siblings) on the path from the abnormal node to the base station would send attestation messages. If sensor nodes are uniformly deployed in two dimensional space, the length of such path would be $O(\sqrt{n})$ [10].

The total number of hops all the attestation messages travel can be estimated as follows, where d is the maximum node degree and PL is the path length.

We assume that each attestation message is unicasted but takes a shortest path to the base station. In other words, the number of hops a message travels is proportional to the distance from the sender to the base station. Such a shortest-path routing is possible using many existing routing algorithms. Note that there may be many shortest paths between a node and the base station if the network is sufficiently dense [9]

$$\begin{aligned} \text{Overhead}_{\text{attest}} &= d \cdot (1 + 2 + \dots + PL) \\ &= d \cdot \sum_{i=1}^{PL} i \\ &= O(PL^2) \\ &= O(n) \end{aligned}$$

The attestation phase takes $O(n)$ number of hops. Therefore the total communication overhead of an aggregation session remains $O(n)$.

In table III, we compare overhead of our work and overhead of related work. Our work has minimum communication overhead as same as most of the related work but our work is able to identify the adversary node as well. This is the prominent point of our work.

VII. CONCLUSION

This paper presented a securing secure data aggregation protocol to defense against false data injection in wireless sensor networks. One of the main contributions is to provide a mechanism to identify the adversary node.

Our solution is lightweight in terms of both computation and communication overhead because it uses MAC which is merely a minimal cryptographic technique and takes $O(n)$ time for both data aggregation and tracing back an attacker. Our solution covers to detect the false data injection during the

TABLE III
COMPARISON OF OVERHEADS

Aggregation Protocols	Overhead		Aggregation Topologies
	Communication	Computation	
Madden [4]	$O(n)$	N/A	Tree
Przydatek [1]	$O(l^2)$, $l < n$	$O(1)$	Centralized
Wagner [2]	$O(n^2)$	N/A	Centralized
Yang [3]	$O(n)$	N/A	Tree
Chan [5]	$O(n)$	N/A	Tree
Vu [6]	$O(n)$	N/A	Centralized
He [7]	$O(n)$	N/A	Tree
Mlahi [8]	$O(n \cdot \ln(n))$	N/A	Tree
Our work	$O(n)$	N/A	Tree

data aggregation and to identify the adversary node even he lies or does not reveal himself where and/or who he is after injecting false data.

Our solution, however, has limitations. It works on the aggregation tree where each sensor node has unique id only. Moreover, it requires sensor nodes (external nodes) that are only dedicated for monitoring the data sensing nodes (internal nodes). Our future work includes designing of reducing the number of external nodes in order to reduce cost of monitoring nodes.

REFERENCES

- [1] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks." In Proceedings of the 1st international conference on SenSys' 03, ACM, 2003, pp. 255–265.
- [2] D. Wagner, "Resilient aggregation in sensor networks." ACM Workshop on Security of Ad Hoc and Sensor Networks, 2004, pp. 78–87.
- [3] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks." the Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing(MobiHoc'06), 2006, pp. 356–367.
- [4] S. Madden, Michael, J. Franklin, and J. M. Hellerstein, "TAG: a Tiny AGgregation service for ad-hoc sensor networks." The 5th Annual Symposium on Operating Systems Design and Implementation (OSDI'02), 2002, pp. 1–16.
- [5] H. Chan, A. Perrig, and D. Song, "Secure hierarchical in-network aggregation in sensor networks." Proceedings of the 13th ACM conference on Computer and Communications Security(CCS'06), 2006, pp. 278–287.
- [6] H. Vu, N. Mittal, and S. Venkatesan, "THIS: THreshold security for Information aggregation in Sensor networks." International Conference on Information Technology(ITNG'07),IEEE, 2007, pp. 89–95.
- [7] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks." IEEE INFOCOM 2007 proceedings, 2007, pp. 2045–2053.
- [8] E. Mlahi and S. A. Aly, "Secure Hop-by-Hop Aggregation of End-to-End Concealed Data in Wireless Sensor Networks." IEEE INFOCOM Workshops, 2008, pp. 1–6.
- [9] S. Zou, I. Nikolaidis, and J. J. Harms, "ENCAST: EnergyCritical Node Aware Spanning Tree for Sensor Networks." Communication Networks and Services Research Conference, 2005, 2005, pp. 249–254.
- [10] H. Chan and A. Perrig, "PIKE: Peer Intermediaries for Key Establishment in Sensor Network." IEEE INFOCOM 2005 proceedings, 2005, pp. 524–535.
- [11] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks." ACM SIGMOBILE Proceedings of the 7th annual international conference on Mobile computing and networking, 2001, pp. 189–199.
- [12] J. R. Taylor, *An Introduction to Error Analysis*. 2nd edition. Sausalito, California: University Science Books, pp 166–8, 1997.
- [13] T. M. Padmaja, N. Dhulipalla, R. S. Bapi, and P. Krishna, "Unbalanced Data Classification Using *extreme outlier* Elimination and Sampling Techniques for Fraud Detection." 15th International Conference on Advanced Computing and Communications, 2007, pp. 511 – 516.

- [14] S. Ross, "Peirce's Criterion for the Elimination of Suspect Experimental Data." Journal of Engineering Technology, fall, 2003.
- [15] H. Alzaid, E. Foo, and J. G. Nieto, "Secure Data Aggregation in Wireless Sensor Network: a survey." 6th Australasian Information Security Conference, AISC, 2008, pp. 93 – 105.