Chapter 48
Distributed Audit Secure Data Aggregation for Wireless Sensor Networks

Zhengdao Zhang and Zhiping Zhou

Abstract Data aggregation can reduce the communication overhead and energy expenditure of sensor nodes, as well as extend the life-cycle of the wireless sensor network. However, because individual sensors may be compromised, the data aggregation also introduces some risks including the false data injection attacks. This paper proposes a distributed audit secure data aggregation protocol. The aggregates are audited at the next level nodes of the aggregators. The communication overload, which Base Station (BS) originates in the attest process, can be avoided. Furthermore, because we can find the false data in the lower level, it is easier to strike out the false data, and only a little fraction of readings are dropped off. To do these, the aggregators attach multi-certificates to the aggregates. Those certificates may include the maximum, minimum, mean reads and those nodes' identifiers. To further reduce the communication overload, we use the watermark method to embed the multi-certificates in authentication part of aggregates. The length of message is kept as same as that under the normal hop-by-hop aggregation protocol with MACs. The analysis shows that our protocol is efficient and provides certain assurance on the trustworthiness of the aggregation result.

Keywords Data aggregation · Distribution · Probabilistic grouping · Watermark · Sensor network security

48.1 Introduction

Wireless sensor networks are usually consist of hundreds or thousands of inexpensive, low-powered sensing devices with limited memory, computational, and communication resources [1]. Due to the low deployment cost requirement of

Z. Zhang (✉) · Z. Zhou
Engineering Research Centre of IOTs Technology and Application, Ministry of Education, Jiangnan University, Wuxi, Jiangsu, China
e-mail: wxzzd.dr@gmail.com

Z. Sun and Z. Deng (eds.), *Proceedings of 2013 Chinese Intelligent Automation Conference*, Lecture Notes in Electrical Engineering 256,
DOI: 10.1007/978-3-642-38466-0_48,

wireless sensor networks, sensors only have limited computation and communication resources. To reduce the communication cost and energy expenditure in data collection, many data aggregation protocols have been proposed. The data aggregation reduces the number of data transmissions, thereby improving the bandwidth and energy utilization in the network. On the other hand, because the raw data will be aggregated at aggregators, data aggregation introduces a lot of security vulnerabilities. For example, the BS cannot authenticate the raw data and find the compromised nodes in network. More seriously, the compromised aggregators can inject an arbitrary false fusion result to make the final aggregation result to far deviate from the true measurement. A lot of researchers focus on the secure data aggregation protocol [2, 3], such as SDA [4], SDAP [5], RSDA [6] and so on. But, the existing methods have some common drawbacks. First, all of them need the BS to attest all suspicious, that introduces a heavy communication overload. Second, if an aggregation result is verified as an attack, the BS drops off this value as well as all reads between attack injection point and BS. This introduces an extra waste of sensor ability.

To solve these problems, a distributed audit secure data aggregation protocol based on the principles of a distributed and cooperative attestation is proposed. A random grouping technique is used to construct a tree topology and logical groups of nodes. The leader node generates an aggregation result. The sibling nodes and the parent node of that leader identify the suspicious reads based on the set of results and a summary of historic reads. After that, the suspicious node in that attestation process proves the correctness of its read. Finally, each group aggregate the result which passes attestation reply to the upper leader and participates in the next aggregation until the result reaches the BS. The remainder of this paper is organized as follows. Section 48.2 describes our data aggregation protocol composed of network model, grouping, aggregation and audit. Security analysis and performance evaluation of our scheme are presented in Sect. 48.3. After that, we summarize our work in Sect. 48.4.

48.2 The Security Data Aggregation Protocol

We assume the BS has unlimited energy, and cannot be compromised. Also, there is a topological tree rooted at the BS which the shape and the distance (in number of hops) between every node are unknown. The transmission mechanism in the protocol is reliable. We also assume there is a broadcast protocol which can provide global broadcast authentication. Every sensor node shares their individual public key with each other. The aggregation tree is constructed as [5]. The count number of y is named C_v.

In our method, the credentials including IDs and Readings will be embedded into a cover message which is calculated based on the message authentication code (MAC) of sensitive data. These credentials will be extracted in the audit procedure by other nodes. The credentials have the format like $(ID_i|R_i)$ The cover messages

always like $MAC(K, ID_i|flag|R_i)$ which is a MAC calculated with the key K. Let $len(.)$ denote the length of data. The total bits of sensitive elements should be less than that of cover data, $\sum len(ID_i|R_i) < MAC(K, ID_i|flag|R_i)$. The details of algorithms can be found in [7].

48.2.1 Subsection Query Dissemination and Leader Selection

We choose the leader nodes as the SDAP [5] does. An example of an aggregation tree and the result of probabilistic grouping are showed in Fig. 48.1. BS first decides the aggregation function F_{agg} and randomly chooses a number $K_{S,BS}$. The BS broadcasts a packet as $BS \rightarrow\rightarrow * : ID_{BS}, K_{S,BS}, F_{agg}$. When a leader node x receives query packet from its parent node y, x stores the F_{agg} and $K_{S,Z}$. After that, x uses a one-way function H to calculate the key for its group as $K_{S,X} = H(K_{S,Z}, ID_X \bmod C_X)$. So, x broadcasts query packet to its children nodes alone the tree as $X \rightarrow\rightarrow * : ID_X, K_{S,X}, F_{agg}$. When an intermediate node receives a query from its parent node, it stores F_{agg} and $K_{S,X}$of the group and broadcasts the query in same way [8].

48.2.2 Aggregation Commitment

Each aggregation packet contains two parts. One is the packet header contained a sender's ID and a flag which implicates different duty of the packet. Another is the data part which contains node's IDs and node's sensed data. We use MAC as a

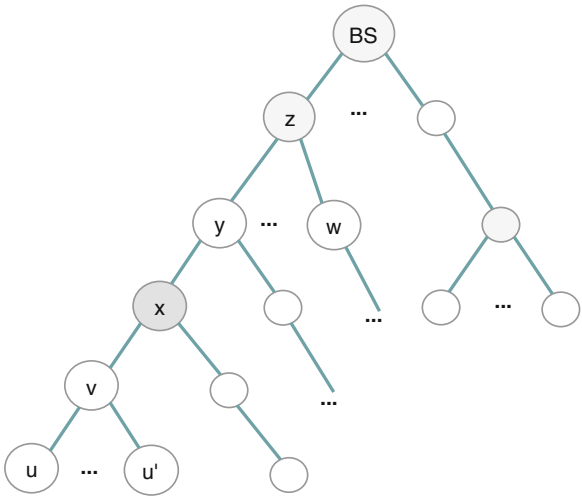

Fig. 48.1 An example of the aggregation tree

cover message, and we embed the node's IDs and the node's sensed data in it [9]. In the left part, we use *WM(key, cover message, hiding data)* to denote the data part.

The aggregation process starts from the leaf nodes towards the BS. The leaf node sends its ID, data to its parent. The packet sent to v from u is:

$$u \rightarrow v : ID_u, flag = 0, C_u = 1, WM\big(K_{S,X}, MAC\big(K_{s,X}, ID_u|flag|R_u\big), (ID_u|R_u)\big)$$

where the R_u is the sensed data of node u. The flag value '0' means this packet doesn't need to be audited, $C_u = 1$ means there is only one data part, and $MAC\big(K_{S,X}, ID_u|flag|R_u\big)$ means the MAC value which is computed with the key $K_{S,X}$.

When a leader node, named x, receives a packet, it first checks whether this packet comes from its child or not. If not, it just drops it. Otherwise, x uses the group key $K_{S,X}$ to extract the hidden data from the data part of the packet. Then, x calculates the MAC with $K_{S,X}$ again, and compares it with the cover message. The packet is accepted only if the authenticity of the MAC is successful.

After obtaining all children's readings $R_1, R_2 \ldots R_{Cx-1}$, x first rearranges all of them with the reading R_X from x in the ascending order of value. Then, x aggregates them as $R_{aggx} = F_{agg}(R_1, R_2, \ldots R_{Cx})$, and uses the key of the upper group to send upward to its parent, named y in Fig. 48.1.

$$x \rightarrow y : ID_x, flag = 1, C_x, WM_1, WM_2$$

$$WM_1 = WM\big(K_{S,Z}, MAC\big(K_{S,Z}, ID_1|flag|R_1\big), (D_1, R_1)\big), \ldots, WM\big(K_{S,Z}, ID_{Cx}|flag|R_{Cx}\big), (ID_{Cx}|R_{Cx})\big)$$

$$WM_2 = WM\big(K_{S,Z}, MAC\big(K_{S,Z}, ID_x|flag|R_{aggx}\big), \big(D_x, R_{aggx}\big)\big).$$

here, flag value '1' means that the values in this packet need to be audited.

When an intermediate node receives a packet from its child node, it then checks the flag. If the flag is '0', it directly forwards the packet to its parent node alone the logic tree. Otherwise, the intermediate node will perform an audit process to verify the correctness of the aggregation value, find the suspicious data, and attest the suspicious data. If all individual sensing data pass the verification, the intermediate node sets the flag value '0', sets the count number $C_x = 1$ again, deletes all individual cover messages (WM_2in data part) and only keeps the last cover message embedded with the aggregation data. Then the intermediate node sends this new packet upward to its parent. For example, in Fig. 48.1, y sends the packet to its parent as follow, if the verification process is successful.

$$y \rightarrow * : ID_x, flag = 0, C_x = 1, WM\big(K_{S,Z}, MAC\big(K_{S,Z}, ID_x|flag|R_{aggx}\big), \big(D_x, R_{aggx}\big)\big).$$

48.2.3 Data Verification and Audit

After a parent node of the group leader, say y, has received an aggregation messages, it needs to verify the authenticity of the aggregated value. This includes verifying the content of the packet and the authenticity of the leader. First, based on the $K_{S,Z}$, y can extract hiding data from each cover message and get the flag and the count value C_x. Secondly, the BS verifies the count number of leader falls in certain range or not. Based on our group partition way, the count number is no more than $k * k$, where k is the maximum number of a logic group. If the count number is large, we have more reasons to suspect that the leader is performing count changing attack. So, the probability for individual audit should be increased.

Choose two cluster centers B_1, B_2 and $B_0 = (B_1 + B_2)/2$ as three references. Let $r = (B_2 - B_1)/4$ as a radius and count the reading which falls in the different regions around them. If no attack is lunched, these references are similar. Otherwise, if the count number of the region around B_0 is much smaller than others, we can suspect one of clusters is forged. If the adverse injects the false readings that gradually deviate from the normal value, the distance between two centers $d = (B_2 - B_1)$ is large. We also can find the suspects easily.

In two situations, the node will ask BS to help perform an audit. One is the parent of a leader suspects that there are some nodes were compromised; another is sibling nodes of leader decide to audit some individual nodes with a per-set probability P_C. To the first situation, the node that has the minimum reading or maximum reading will be audited at first. Then, if anyone fails the audit, all nodes whose reading falls into the same cluster with failed node will be audited. The parent node y sends an audit request to ID_1 with the public key K'_{ID_1} of ID_i as

$$y \rightarrow ID_i : ID_y, flag = 3, WM\left(K'_{ID_1}, MAC\left(K'_{ID_1}, flag|ID_y|K'_y\right), \left(ID_y|K'_y\right)\right)$$

D_i then uses y's public key K'_y to response this request as

$$ID_i \rightarrow y : ID_i, flag = 3, WM\left(K'_y, MAC\left(K'_y, flag|ID_i|R_i\right), (ID_i|R_i)\right)$$

After receiving R_i, y compares it with the reading extracted from the leader. If these two readings are same, the node ID_1 passes the audit. These readings will be transmitted to the group leader as an individual aggregation result. They will be accepted only other readings or aggregation results in this group can support it. If two readings are different, we know the leader or the ID_i was compromised. So, y will report the compromise to its group leader z. Supposed that ID_1 was failed in audit. Y recalculates the aggregation result as $R'_{aggx} = F_{agg}(R_1, \ldots, R_{i-1}, R_{i+1}, \ldots R_{Cx})$. Then, Y reports the compromise with the new result.

$$y \rightarrow z : ID_y, flag = 0, C_x = 1, WM\left(K_{sg,z}, MAC\left(K_{sg,z}, ID_x|flag|C_x|R'_{aggx}\right), \left(ID_x|R'_{aggx}\right)\right), WM\left(K_{sg,z}, MAC\left(K_{sg,z}, ID_i|R_i\right), (ID_i|R)\right)$$

The sibling node first extracts all readings from the cover messages with the group key. Then, for each reading, sibling node performs an audition procedure based on a per-set probability named P_C. So, the sibling node uses a random number generator to get a random number. If the number is less than P_C, the current reading will be audited as above. This process will repeat until all readings take part in the audit process. After that, if any conflicts are found, the sibling node also reports them to the group leader. In Fig. 48.1, node w acts as a sibling node and it will report the compromise as:

$$w \rightarrow z : ID_w, flag = 4, C_p WM\left(K_{sgg.z}, MAC\left(K_{sg.z}, flag|ID_w|R_w\right), \left(ID_w|R_w\right)\right).$$

48.3 Security Analyses and Performance

In our method, BS always disseminates aggregation query with a different random number $K_{s,SB}$ and the number acts as a key as well. Based on this original key, every group produces a different key for every round too. If the malicious node replies past reading again, the leader node and the other sibling nodes can't extract the correct hiding data with a new group key and will get a different MAC. Then, this packet will fail in the authentication and be discarded.

An attacker cannot selectively compromise some nodes and makes some of them to appear in same group, because the groups are decided by a random process. A compromised node may change a real value corresponding to an abnormal event to a normal value. This attack, named potential event suppression attack, is a big vulnerability to the data aggregation. Our method can deal with this attack because the sibling nodes of leader will audit every reading with a probability. If the leader gets rid of the real abnormal node from its children list, the parent node can easily find that by checking the leader selection condition. Our method mitigates the count changing attack because (1) every reading taking part in aggregation will be audited with same probability, and (2) the *ID* of every node is unique and connected with a pair of keys which is stored at BS.

In the group, every leaf node only can report a reading with count number '1'. So, if the attacker launches a count changing attack, the compromised node must be a group leader node. Then, to the each sibling node of that leader node, it will audit every reading with a per-set probability P_C. If the compromised node forges a real *ID* and the compromised node can't get the key pair of that *ID*, the sibling node asks for audit information, the real node responses that query. Then the sibling node will easily find the attack because the two readings are different.

A compromised node also may forge small count but extreme data to modify the final aggregation result. In this situation, the attacker also needs to compromise a group leader node, and then forge the readings of its children nodes. This is as same as the second case of count changing attack, and can be detected as the same way.

For simplicity, we only consider the case where the leaf nodes transmit their readings and no readings are expected from aggregator nodes. We assume a general tree hierarchy in which every node has b children and the depth of the tree is d as same as in [2]. Assume the length in bits of reading from the leaf nodes is x. The sensor node ID in bits will be denoted as y. Let the bits of count number equals 8. Also, we denote the MAC's length in bits as z. Since TinyOS packet has the maximum size of 36 byte, including 29 byte payload and 6 byte header, we denote header as oh to compute the overhead bits transmitted within the network.

We assume each group has an average size of s, so the number of the groups is $(N/s) + 1$. Also, the distance from each leader to the BS can be considered as $d/2$. If a node sends its *ID*, three bits of flag, a count number and a MAC, the length is $y + 3+8 + z+oh$. The total number of bits in aggregation is approximated by

$$\left(\frac{b^d - b}{b-1}\right)(s-1)(y+z+11+oh) + \left(\frac{N}{s}\right)(s-1)z$$

And, the overhead for attestation is around

$$\left(\frac{b^{d-1} - b}{b-1}\right)\left(2(s-2)\left(\frac{N}{s}\right)\left(\frac{d}{2}\right)+1\right)(y+z+3+oh)$$

Our protocol has the same communication overload in aggregation procedure as SDAP does. In the attestation procedure, our protocol has fewer communication overloads if there are several attested groups. Based on the same simulation assumption with [5], every attestation procedure of SDAP need relay no less than *88 packet*hop*. If there are n attested groups, the relay is no less than *88*n packets*hop*. Our protocol totally need relay $95*30*P_c$ packets*hop for any number of suspect groups. Notice that the P_c is a small number in our protocol, E.g. 0.1. Then, our protocol relay fewer packet*hop if the number of attested group is more than 3. If the number of attested group is 10, our protocol only need relay *32.39 % packet*hop* of SDAP.

48.4 Conclusion

In this paper, we propose a Distributed Audit Secure Data Aggregation Protocol for large-scale sensor networks. By using the probabilistic grouping method, we partition the aggregation tree into the different logic groups. The upper level nodes can also check the grouping results to prevent some malicious nodes to compromise the special group leaders. A cluster procedure is performed at aggregator to find any suspect readings, and an audit procedure is performed in the next upper group of aggregator to audit the aggregation result. To simplify the audit process, we attach multi-certificate in the aggregates, and use watermark algorithm to keep the message length no longer than the message length in other protocol. In the future, the potential of integrating with CDMA watermark algorithm will be investigated too.

Acknowledgments This work is supported by Fundamental Research Funds for Central Universities, Chinese Department of Education, under Grant JUSRP111A49.

References

1. Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. Comput Netw 52(12):2292–2330
2. Hani A, Ernest F, Juan M, Gonzalez N (2008). Secure data aggregation in wireless sensor network: a survey. In: 6th Australasian information security conference, vol 81 of CRPIT, Wollongong, NSW, Australia, pp 93–105
3. Ozdemir S, Xiao Y (2009) Secure data aggregation in wireless sensor networks: a comprehensive overview. Comput Netw 53(12):2022–2037
4. Hu L, Evans D (2003) Secure aggregation for wireless network. In: SAINT workshops, IEEE Computer Society, pp 384–394
5. Yang Y, Wang X, Zhu S et al (2008) SDAP: a secure hop-by-hop data aggregation protocol for sensor networks. ACM Trans Inf Syst Secur 11(4):1–43
6. Alzaid H, Foo E, Nieto G (2008) RSDA: Reputation-based secure data aggregation in wireless sensor networks. In: Proceedings of the 1st international workshop on sensor networks and ambient intelligence, Dunedin, New Zealand
7. Yang J, Sun X, Wang B, Xiao X (2010) Bloom filter-based data hiding algorithm in wireless sensor networks. In: The 5th international conference on future information technology, Busan, Korea, pp 1–6
8. Perrig A, Szewczyk R, Wen V, Culler D, Tygar J (2001) SPINS: security protocols for sensor netowrks. In: Mobile computing and networking, pp 189–199
9. Zhang W, Liu Y, Das SK, De P (2008) Secure data aggregation in wireless sensor networks: a watermark based authentication supportive approach. Pervasive Mob Comput 4(5):658–680