Performance Of Data Aggregation Using Anonymous Aggregator Election In Wireless Sensor Networks

Dr. A.Manikandan[1], Dr. V.Ganeshbabu[2] and P.Sakthivel[3]
[1]Asso. Prof & Principal, Muthayammal Memorial College of Arts & Science, Rasipuram, Tamilnadu, India.
[2]Asst. Prof, Dept. of CS, Govt. College for Women, Maddur, Karnataka, India.
[3]Asst. Prof, Dept. of CS, Muthayammal Memorial College of Arts & Science, Rasipuram, Tamilnadu, India.

Abstract: Wireless sensor and actuator networks are potentially useful building blocks for cyber-physical systems. It must typically guarantee high-confidence operation and makes strong requirements of building blocks. Here, the requirement of dependability means resistance against both accidental failures and intentional attacks. It should be addressed at all layers of the network architecture, including the networking protocols and the distributed services built on top of them, as well as the hardware and software architecture of the sensor and actuator nodes themselves. In this paper, we focus on the security aspects of aggregator node election and data aggregation protocols in wireless sensor networks (WSN).

Keywords: vehicular networks, communication, announFirst, aggregator, cluster and protocols.

1. Introduction

Data aggregation in WSN helps to improve the network of the scalability and energy efficiency that it introduces some security issues[1]. The aggregator nodes accumulate and hoard aggregated sensor readings and correspond with base station are smart targets of physical node destruction & jamming attacks. Our proposal of new private aggregator node election protocol is resistant to internal attacks originating from compromised nodes. We proposed a new private data aggregation protocol which preserved the anonymity of the aggregator nodes during the data aggregation process. In private aggregator node election protocol, each node chooses locally in a probabilistic way to suit an aggregator or not an aggregator. Also the nodes execute an anonymous veto protocol to confirm if at least one node became aggregator. The protocols used to protect sensor network applications that rely on data aggregation in clusters, and where locating & disabling the aggregator nodes are highly undesirable. In physical area, sensor network organized into clusters and use in-network data aggregation in order to make sure scalability and energy efficient process.

1.1. System models and attacker models

Sensor network of sensor nodes communicate with each other via wireless channels [2]. Each node communicates with the nodes within its radio range. These kinds of nodes are called neighbors of the node. In order to communicate with distant nodes, the nodes use multi-hop communications. Event driven networks can be used for reporting special usually dangerous but infrequent events like fire in a building. There is no need of clustering and data aggregation in event based systems, thus private cluster aggregator election and data aggregation is not applicable there. The last one of the query driven network of operator sends a query to the network and it sends a response. The nodes may be aware of their geographical locations, and they may already be partitioned into well cleared geographical regions. These regions are the clusters and the objective of aggregator election protocol is to elect an aggregator within each geographical region. The goal of the election is to elect one node in every preset cluster. Any node may announce itself as an aggregator, the nodes within a certain number of hops on the topology graph may join that node as cluster members. The location and topology based approaches are shown in Figure 1.1.

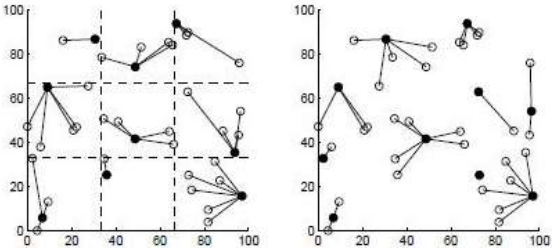


Figure 1.1 Result of a location based and topology based (right) one-hop aggregator election protocol.

This figure of solid dots represents the aggregators and empty circles represent cluster members. In case of location based or preset clustering, the scope of a flood is restricted to a given geographic region or preset cluster[3]. Nodes within that region rebroadcast the message to be flooded when they receive it for initial period. Nodes are having different preset cluster IDs simply drop the message. In case of topology based clustering, it is assumed that the broadcast messages have a Time-to-Live field that controls the scope of the flooding. Any node that receives a broadcast message with a positive TTL value for the first time will automatically decrement the TTL value and retransmit the message. Duplicates and messages with TTL smaller than or equal to zero are silently discarded. In the location based case, the cluster peers of *S* are the nodes. The main functional requirement of any clustering algorithm is that either node *S* or at least one of the cluster peers of *S* will be elected as aggregator. The leader of each cluster is called cluster aggregator, or simply aggregator. An attacker who wants to discover the identity of the aggregators can eavesdrop the communication between any nodes, can actively participate in the communication and can physically compromise some of the nodes. A compromised node is under the full control of the attacker, the attacker can fully review the inner state of that node, and can control the messages sent by that node. Compromising a node is a much harder challenge for an attacker than simply eavesdropping the communication [4]. So, we propose two solutions. The first protocol can completely withstand a passive eavesdropper. The second advanced protocol can withstand a compromising attacker, with only leaking information about the compromised nodes. In case of a passive adversary, a rather easy solution could be based on a general shared global key. Using this pseudo random number generator, every node can construct locally the same pseudo randomly ordered list of all nodes. These lists will be identical for every node because all nodes use the similar seed and the similar pseudo random number generator.

2. Basic protocol

In private aggregator node election, each node starts executing the protocol roughly at the same time. The protocol terminates after a predefined fix amount of time[5]. During this process, node didn't receive aggregator announcement. It broadcasts an aggregator announcement message as a cluster aggregator. This message is transmit among the cluster peers of the node sending the announcement. All messages sent in the protocol are encrypted such that only the nodes to whom they are intended and decrypt them. It is assumed that each node shares a common key with all of its cluster peers[6]. In addition, in order to avoid that message originators are identified as cluster aggregators, the nodes are required to transfer dummy messages that cannot be discriminated from the announcements by the external observer. In such a protocol, the probability that neighboring nodes share a common key is high and unused keys are deleted. The central method for exchanging a cluster key with the neighboring nodes is to transfer the similar random key to each neighbor encrypted. The protocol consists of two rounds, where the length of each round is τ. The nodes are synchronized, they all identify when the first round begins, and what the value of τ is. Initially, each node starts two random timers, T1 & T2, where T1 expires in the first round and T2 expires in the next round. Each node initializes a binary variable, called announFirst that determines in which round the node interest to transfer a cluster aggregator announcement.

Algorithm 1. Private cluster aggregator election algorithm

start T1, expires in rand(0,τ)
start T2, expires in rand(τ ,2τ)
announFirst = (rand(0,1) $\leq \gamma$)
CAID = -1 // ID of cluster aggregator of node
while T1 NOT expired **do**
if receive ENC(announcement) AND (CAID = -1) **then**

CAID = ID of sender of announcement
end if
end while
if announFirst AND (CAID = -1) **then**
broadcast ENC(announcement);
CAID = ID of node itself;
else
broadcast ENC(dummy);
end if
while T2 NOT expired **do**
if receive ENC(announcement) AND (CAID = -1) **then**
CAID = ID of sender of announcement
end if
end while
if (NOT announFirst) AND (CAID = -1) **then**
broadcast ENC(announcement);
CAID = ID of node itself;
else
broadcast ENC(dummy);
end if

Table 2.1. Estimate time of the building blocks on a MICAz

Algorithm	**Generation** [ms]	**Verification** [ms]
SHA-1 [Ganesan et al., 2003]	1.4	-
RSA 1024 bit [Piotrowski et al., 2006]	12040	470
RC4 [Ganesan et.al.,2003]	0.1	0.1
RC5 [Ganesan et.al., 2003]	0.4	0.4

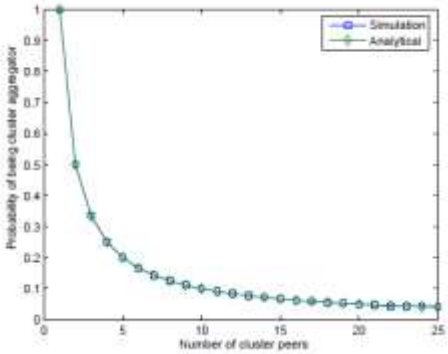
The probability of announFirst is set to the first round is γ, which is a system parameter. In the first round, every node *S* waits for its first timer T1 to expire. If *S* accepts an announcement before T1 expires, then the sender of the announcement will be the cluster aggregator of *S*. When T1 expires, *S* broadcasts a message as follows: if announFirst is set to the first round and *S* has not received any announcement yet, then *S* sends an announcement, in which it announces itself as a cluster aggregator. Otherwise, *S* transfers a dummy message. In both cases, the message is encrypted such that only the cluster peers of *S* can decrypt it. The second round is similar to the first round. When T2 expires *S* transmits a message as follows: if announFirst is set to second round and *S* has not received any announcement yet, then *S* transfers an announcement, otherwise, *S* sends a dummy message. In both cases, the message is encrypted. If a node is compromised, the adversary learns only the identity of the cluster aggregators whose announcements have been received by the compromised node[7]. In WSNs, it must be analyzed what happens if some messages are delayed/lost in the noisy unreliable channel. If an announcement is delayed/not delivered to a node, then the recipient will not choose the sender as cluster aggregator. It will choose a node who sent the announcement later or the node elects itself and sends an announcement.

2.1. Protocol analysis

The main goal of the attacker is to disclose the identity of cluster aggregators[8]. So, the attacker can eavesdrop, modify, and delete messages and capture some nodes. Initially the logical attacks analyzed where the attacker does not detain any nodes, then the results of a node detain. As all the inter node communication is encrypted and authenticated, it cannot get any information from the messages themselves, but it can get some side information from simple traffic and topology analysis. The model of density based attack is an external observer cannot trivially identify the cluster aggregators. However it can still use side information and deduce some nodes to be cluster aggregators with higher probability than some other nodes. Such side information is the number of the cluster peers of the nodes. This number associates with the local density of the nodes that is why this attack is called density based attack. The probability of cluster aggregator depends on the number of the cluster peers of the node. On the other hand, if the node has a

larger number of cluster peers, then the probability of receiving an announcement from a cluster peer is large and the probability that the node itself becomes cluster aggregator is small. The number of cluster peers can be deduced from the topology of the network, which may be known to the adversary[9]. The probability of becoming a cluster aggregator is approximately inversely proportional to the number of cluster peers:

$$Pr(CA(S)) = 1\ D(S) \quad \text{----------} \quad (2.1)$$



where CA(S) is the event of S being elected cluster aggregator and D(S) is the number of cluster peers of node S. Equation 2.1 is quite sharp, it is nearly close to the simulated results. There are two approaches are used to mitigate this problem. One is to take the number of cluster peers of the nodes into account when generating the random timers for the protocol. The second is to balance the logical network topology that every node has the same number of cluster peers. The first approach can be the fine tuning of the distributions. The coefficients of the polynomial are set as resulting curve is the closest to uniform distribution. Actually by modifying γ, the other attack discussed in the next section can be mitigated, so here we propose a solution which does not set the γ parameter. The second approach modifies the number of cluster peers of a node to reach a common value.

Let us denote this value by α. An efficient approach to diminish this problem is to change the number of cluster peers such that it considers a common value α for all of them. In theory, this value can be anything between 1 and the total number N of the nodes in the network. In practice, it should be around the average number of cluster peers, which can be estimated locally by the nodes[10]. For example, assuming one-hop communications the following formula can be used:

$$\alpha = (N - 1)\ R2\pi\ A + 1\ \ E(D(S)) \quad \text{------} \quad (2.2)$$

In the following figure, R is the radio range, and A is the size of the total area of the network. The formula is based on the fact that the no. of cluster peers is proportional to the ratio between radio coverage & total area. If a node S has more α cluster peers it discard the messages from D(S) – α randomly chosen cluster peers. If S has less than α cluster peers it must get new cluster peers by the help of its actual cluster peers.

Figure 2.1. Probability of being cluster aggregator as a function of the number of cluster peers.

The advanced private data aggregation protocol consists of four main parts. The first part of initialization provides the required communication channel[11]. The second part is needed for data aggregator election. The third part is needed for data aggregation. The last part must support the queries, where an operator queries some stored aggregated data.

Table 2.2. Complexity of the advanced protocol.

Protocol Usage	Election	Aggregation	Query
Message Complexity	0 (N^2)	0 (N)	0 (N)
Modular exponentiations	$4N^1$	0	0
Hash computations	0	0	1

Here, N is the number of nodes in the Cluster. The initialization phase is responsible for providing the medium for authenticated broadcast communication. Each node has some unique cryptographic credentials to enable authentication. Each message contains the cluster identifier and addressed to a cluster different from one node belongs to discard by the other node. These methods are in two types. Initial

method of transmit authentication enables a sender to broadcast some authenticated messages efficiently to a large number of potential receivers. One solution to the node compromise is the hop by hop authentication of the packets. The second method of broadcast communication enables transferring information from one source to every other network. In vehicular networks, it can be implemented flooding network/unicast messages. The value of transmit communication is hiding the traffic patterns. An efficient way of this vehicular network is the usage of connected dominating set (CDS). This set S of graph G is definite as a subset of G such that every vertex in G−S is adjacent to at least one member of S, and S is connected.

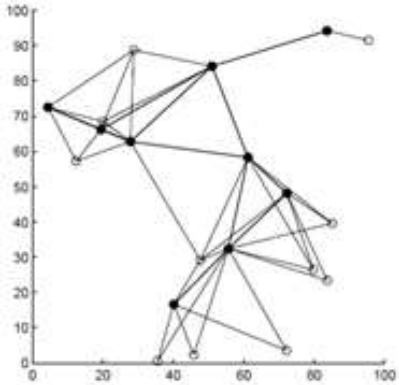


Figure 2.2. Connected dominating set. Solid dots represent the dominating set and empty circles represent the remaining nodes.

3. Data Aggregator Election

The goal of aggregator node election protocol is to elect a node that can hoard the measurements of entire cluster. The election is successful if at least one node is elected. The election process consists of 2 main steps. i) Each node decides, whether it wants to be an aggregator, based on some random values. ii) Anonymous veto protocol is run, which reveals only the information. In step(i), every node elects itself aggregator with a given probability p and result of election is kept secret, participants only want to know that the no. c of aggregators is not 0, without illuminating the identity of cluster aggregators. We denote the random variable representing no. of elected aggregators with C. It is easy to see that the distribution of C is binomial:

$$Pr(C = c) = (N\ c\ )\ pc\ (1 - p)N-c \text{ ------ } (2.3)$$

The expected no. of aggregators after the step (i) is: cE = Np. The probability that no cluster aggregator is elected is: (1 − p) N. To avoid this situation when no node is elected, nodes must run step (ii) and at least one node is elected as aggregator node. This problem can be solved by an anonymous veto protocol. The secrecy of election subprotocol depends on the parts of protocol[12]. The random number generation does not leak any information about the identity of the aggregator nodes, if the random number generator is safe. A cryptographically secure random number generator called TinyRNG. The message complexity of election is O(N2), which is acceptable as the election is run rarely. In vehicular networks, the links in general are not reliable, packet losses occur in time to time. After the election subprotocol, every node is equiprobably aggregator node. The election subprotocol ensures that at least one aggregator is elected and this node(s) is aware of its status. An outside attacker does not identify the identity of the aggregators or even actual number of the elected aggregator nodes. An attacker, who compromised one or more nodes, can decide whether the compromised nodes are aggregators.

Misbehaving nodes

The election process of compromised node is only elected aggregator, because a compromised node may not store the aggregated values. During the aggregation, a misbehaving node can modify its readings. The most interesting subprotocol from the perspective of misbehaving nodes is the query protocol. Here, a compromised node can simply modify the result of query in the following way. A compromised node can add an arbitrary number X to the hash instead of using 0 or M. It is easy to see, that if X is selected from interval $[A,B]$, then after subtracting the hashes, the resulting sum R' will be an integer in interval $[(c+1)A, (c+1)B]$. A compromised node can further increase its influence by choosing X from the interval $[iA, iB]$. The resulting sum R' will be in the interval $[(c + i)A, (ci)B]$. If X is not selected from interval $[jA, jB]$, $j = 1 . . .N$, then result can be outside of decidable intervals. If result is in a legitimate interval ($\exists j, R' \in [jA, jB]$),

then the operator can further check the consistency by calculating R' mod j. If the result is 0, then no one misbehaving node in the network. If the result is non 0, then a misbehaving node is appear in the network. It is tough for the attacker to guess j, because it neither knows the actual no. of aggregators, nor can calculate R' from R by subtracting unknown hashes. If the modulus is 0, it can further test the cluster for misbehaving nodes with the help of aggregated bit in the queries. If the 2 sums are equal, then the operator check the results from the second round. If values are equal, then no misbehavior detected, otherwise some node(s) misbehave in cluster. This algorithm only detects if some misbehavior is occurred in cluster, but does not find the misbehaving node.

4. Related Work

There are 2 main groups are classified in privacy protection techniques: data-oriented & context oriented protection. In data-oriented protection, confidentiality of measured data must be sealed. In context oriented protection covers the location privacy of the source & base station(BS). The source location privacy is a problem in event driven networks, where the existence and location of the event is the information, which must be hidden. The main difference between hiding the BS and the network aggregators is that a WSN frequently contains only one BS which is a predefined node, while at the same time there are more in network aggregators used in one network, and the nodes used as aggregators are periodically changed. The problem of private cluster aggregator election in WSN's is strongly associated to anonym routing in WSNs. Here, anonym routing supports any traffic pattern and generally handles external attackers.

5. Conclusion

In this paper, we proposed 2 private aggregator node election protocols for WSN that hide the elected aggregator nodes from attacker. It cannot locate and disable them. Basic protocol provides fewer guarantees than advanced protocol. Advanced protocol hides the identity of the elected aggregator nodes even from insider attackers, thus it handles node compromise attacks too. The aggregation protocol allows the aggregator nodes to collect sensor readings & respond to queries of the operator. In aggregation and query protocols are resistant to both external eavesdroppers and compromised nodes participating in the protocol. The communication in the advanced protocol is based on connected dominating set, which suits to WSN. This algorithm detect misbehave nodes in the query phase. Commonly, protocols amplify the dependability of sensor networks and applied in mission critical sensor network applications for high-confidence cyber-physical systems and actuators.

References

[1] N. K. Prema and Dr. A. Arul S.K, International Research Journal of Computer Science, 2393-9842, Vol 1, Issue 3, pages 63-79, 2016.

[2] M. Abadi & C. Fournet. *Theoretical Computer Science*, 322(3):427–476, 2014.

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2016.

[4] M. Aoki & H. Fujii. Technical issues on vehicle control application. *Communications Magazine, IEEE*, 34(10):90–93, 2017.

[5] A.R. Beresford & F. Stajano. Mix zones: User privacy in locationaware services. *Pervasive Computing & Communications Workshops. Proceedings of the 2nd IEEE Annual Conference on*, pages 127–131. IEEE, 2017.

[6] Z. Berki. *Development of Traffic Models on the basis of Passanger Demand Surveys Thesis of the PhD dissertation*. PhD thesis, Budapest University of Technology and Economics, 2017.

[7] M. Beye & T. Veugen. Cryptology ePrint Archive, Report 2011/395, 2016.

[8] M. Beye and T. Veugen. Anonymity for key-trees with adaptive adversaries. *Security and Privacy in Communication Networks*, pages 409–425, 2016.

[9] Jan Camenisch and Markus Stadler. Proof systems for general statements about discrete logarithms. Technical report, Department of Computer Science, ETH Z¨urich, 2015.

[10] B. Carbunar & Y. Yu, L. Shi, Query privacy in wireless sensor networks. In *Sensor, Mesh & Ad Hoc Communications and Networks, SECON'14. 4th Annual IEEE Communications Society Conference on*, pages 203–212. IEEE,2014.

[11] H. Chan & A. Perrig. Security & privacy in sensor networks. *Computer*, 36(10):103–105, 2016.

[12] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, pages 197–215. IEEE Computer Society, 2016.