Secure and Efficient Data Collection in Sensor Networks

Cristina Cano[1], Manel Guerrero[2], Boris Bellalta[1]
(1) Universitat Pompeu Fabra
(2) Universitat Politecnica de Catalunya
cristina.cano@upf.edu, guerrero@ac.upc.edu, boris.bellalta@upf.edu

***Abstract*— Sensor networks are a very specific type of wireless networks where both security and performance issues need to be solved efficiently in order to avoid manipulations of the sensed data and at the same time minimize the battery energy consumption. This paper proposes an efficient way to perform data collection by grouping the sensors in aggregation zones, allowing the aggregators to process the data generated (sensed) inside the aggregation zone in order to minimize the amount of transmissions to the sink. Moreover, the paper provides a security mechanism based on hash chains to secure data transmissions in networks with low capability sensors and without the requirements of an instantaneous source authentication.**

I. Introduction

Nowadays, the manufacture of inexpensive wireless sensor nodes powered by batteries opens a broad range of applications [1] (from environment observation to health applications, from home to big commercial applications, etc.). Nevertheless, the means (or protocols) by which those sensor networks collect the information they sense in such a way that it will be efficient and that the sensor nodes will not run out of battery to soon are still not in place. Routing protocols for general purpose wireless networks (like AODV [2] and OLSR [3]) cannot be used in wireless sensor networks efficiently due to the specific (unnecessary) characteristics of their control traffic. Moreover, the use of security will be fundamental for some common sensor applications, that typically are implemented with computationally expensive cryptographic primitives.

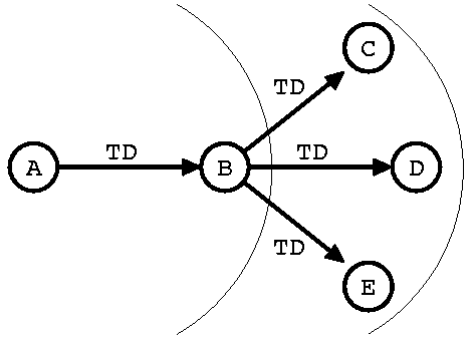
Therefore, the design of security mechanisms for sensor networks has to take into account that sensors are, typically, nodes with very limited memory and computer power. In this paper, we propose the utilization of *hash chains* as a way of providing delayed authentication but satisfying the low computation capabilities of the sensors without requiring any type of key pre-settings. Hash chains have being used as an efficient way to obtain authentication in several approaches that tried to secure routing protocols. For example, in [4], [5] and [6] they use them in order to provide delayed key disclosure. While, in [7], hash chains are used to create one-time signatures that can be verified immediately. The main drawback of all the above approaches is that all of them require clock synchronization. SAODV uses hash chains to authenticate hop counts [8], [9]. In SEAD [10] (by Hu, Johnson and Perrig) hash chains are also used in combination with DSDV-SQ [11] in a very similar way (this time to authenticate both hop counts and sequence numbers). At every given time each node has its own hash chain. The hash chain is divided into segments, elements in a segment are used to secure hop counts in a similar way as it is done in SAODV. The size of the hash chain is determined when it is generated. After using all the elements of the hash chain a new one must be computed. SEAD can be, in theory, used with any suitable authentication and key distribution scheme. But finding such a scheme is not straightforward.

Additionally, a new simple tree discovery and routing through the tree algorithm is proposed. One solution to improve the efficiency in data transmission is the use of aggregation techniques (i.e., join several data packets in a single one in order to reduce the unnecessary overhead transmitted). This can result in lower battery consumption and an increment of capacity (in messages transmitted by the network with the same transmission resources). Krishnamachari et al. analyze, in [12], from a theoretical point of view the benefits and drawbacks of doing data aggregation in wireless sensor networks. Nevertheless, they do not show a way of performing that data aggregation. Kalpakis et al. present, in [13], an algorithm to solve the data collection problem for wireless sensor networks. Nevertheless, the algorithm they provide is polynomial-time. In SIA [14], Przydatek et al. considers information aggregation in sensor networks. But they assume that there are some nodes that perform aggregation and other nodes that do not, and leave how to decide which nodes are aggregators and which are not out of the scope of their paper.

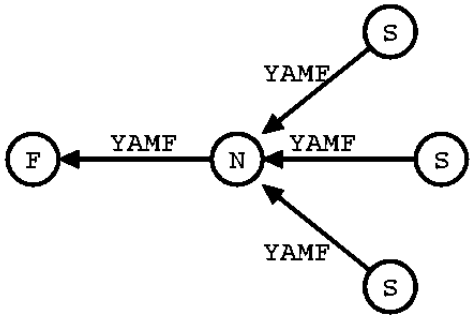
Finally, the new network layer with security and data aggregation is evaluated in order to asses what are the costs of using security in terms of performance and the benefits of using aggregation / data fusion.

II. Building a tree with aggregation zones

In a typical sensor network, there is a 'sink' (a node that collects the information that the other nodes sense) and the sensor nodes. Thus, the streams of information have the structure of a tree that has the sink node as its root, and which is probably one of the most efficient structure for data collection and aggregation for sensor networks. The challenges are, however, the construction or discovery of the tree, the security of the routing protocol and how efficiently the different streams can be merged when converge in the same path.



(a)



(b)

Fig. 1. (a) Propagation of the TD message, (b) Propagation of the YAMF message

A. The Discovery of the Tree

In a sensor network with no or slow node mobility, the discovery of the 'collector tree' will only need to be done after nodes deployment and when the sink notices that it stops receiving an important part of the sensor nodes' reports (which might be due to the dead of one or many sensor nodes or due to a change in the link connectivity). To perform a tree discovery, the sink broadcasts a Tree Discovery ('TD') message. Every node that receives it marks the sender as its father in the tree and unicasts a 'YAMF' (You Are My Father) reply back to the node. Therefore, once the tree discovery is over, every node knows which node is its father and which nodes are its children. Figure 1.(a) shows how TD messages get propagated, and figure 1.(b) shows how YAMF messages travel back.

The 'TD' message will include information of whether the sensor nodes have to report with what they sense periodically, or only as a reaction to certain event.

1) Adopting a Node: In the case that a node detects that it has lost its link connectivity with its tree father, it can broadcast an 'ILMF' (I've Lost My Father) message to its neighbors. A node that receives an 'ILMF' message will forward it towards the sink in the same manner as it would do with a message containing sensed data. When the sink receives the 'ILMF' message or messages it will decide whether to trigger a new tree discovery or not. With this the sink can consider to trigger a new 'TD'.

A node that issues an 'ILMF' message because it has realized that its father is not reachable anymore might set an 'Adoption' flag in the message. This flag indicates that the node is willing to accept a neighbor as its new father. A neighbor receiving an 'ILMF' message will offer to adopt the node and it will issue an 'IWTA' (I'm Willing To Adopt you) message. The orphan node will chose among the nodes who sent him the 'IWTA' (typically it will choose the first one) and will notify it by sending to its future father an 'YMNF' (You're My New Father) message. In the case that no node is willing to adopt the orphan node, the node will issue another 'ILMF', but this time without the 'Adoption' flag set.

B. Aggregation nodes

Among the sensor nodes there are some special nodes which are capable of aggregating data (see Figure 2, where an sketch of the network topology is shown). These nodes can be standard sensor nodes that, for example, with a probability p change to an aggregation role during a certain period of time. However, due to the requirements to authenticate the data received before aggregate it, an special consideration for them is required. These nodes aggregate the data received in a fixed duration time period (D_{agg}) by performing some kind of data fusion (for instance computing the mean). Moreover, they must perform data authentication in a similar way as it would be done at the sink.

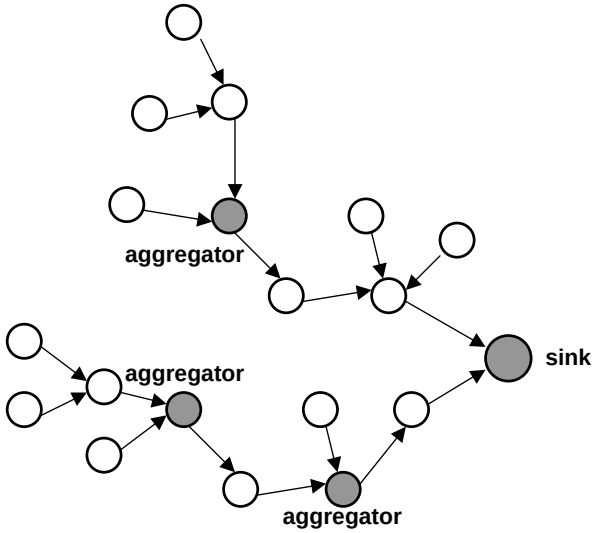


Fig. 2. Sketch of the tree with aggregation nodes scenario

The use of aggregation reduces the transmitted / received bits, and thus, the power consumption is reduced. With respect to the MAC protocol, the use of aggregation can decrease the transmission probability of a sensor to the channel which

reduces as well the collision probability and the overall use of the transmission resources.

III. Solving the Data Authentication

A malicious node must only be able to give wrong data about its sensor data and to decide whether it forwards or not the sensor data it is supposed to forward towards the sink. Both cases are detectable by the aggregation nodes or the sink. Thus, there are two security services which will be guaranteed:

- **Integrity**: The sink needs to be able to verify that the information that it is being reported by a sensor node has not been altered by a forwarding node.
- **Source authentication**: In addition, the sink needs to be able to verify that a node providing the data is the one it claims to be.
- The two last security services combined build **data authentication**.

Availability is outside of the scope of this paper. Although of course it would be desirable, it does not seem to be feasible to prevent denial-of-service attacks in a network that uses wireless technology (where an attacker can focus on the physical layer without bothering to study the routing protocol). Notice that it is assumed that **Confidentiality** is not required.

It is important to look at some possible attacks to the presented protocol and to other protocols designed to be used in sensor networks.

- **Multiple Personality**: In this attack a node pretends to be $\mathbf{n}$ nodes. Every time it is supposed to provide an information it does for all of its multiple personalities. If $\mathbf{n}$ is not too small in comparison to the total number of sensors in the network, then the perception of the situation by the sink will be very misled.
- **Man-in-the-middle**: In this attack the node has multiple personality of as many nodes as sons, grandsons, etc., he has. When it receives the sensed data by them it modifies the data at its will and uses its other personalities to forward the modified sensed data.

A. Using Secret Keys and/or Public Key Cryptography

Those two previous attacks must be taken into account when studying how to implement source authentication. It could be argued that the only way for a sink to detect such attacks, would be to know (in a way) all the sensor nodes or the use of a secret key known between each of the nodes and the sink.

Nevertheless, if the requirement of sharing a secret key between each of the sensor nodes and the sink before network deployment is not a feasible one, an alternative method could be used: The sink would have a key pair and its public key would be known by all the sensor nodes. After network establishment the sensor network would collect the secret key of each of the sensor nodes encrypted with the public key of the sink in the same manner sensed information is collected. The main problems with this approach are that attacks of the kind of 'Multiple personality' are not avoided and that public key cryptography is computationally expensive.

B. Using Hash Chains

If it is acceptable that the sensed data that a node sends is not authenticated immediately, but with the next data transmission, hash chains could be used to obtain delayed authentication. With this technique, the identity of a node gets marked by its TOP-HASH.

This approach will not solve the problem with attacks of the kind of 'Multiple personality' but requires no pre-sharing of any keys, data or whatsoever. On the other hand, it will limit the number of sensed data messages that sensor nodes can send in an authenticated way with the same hash chain, re-computing it (if allowed) when all the hashes has been used. Nevertheless, the number of hashes will, arguably, be higher than the messages that it will be sent before battery deployment for most scenarios.

Each node generates a seed (typically a random number that must be kept in secret) for the hash chain. Then it calculates each of the links of the hash chain and store a subgroup of them in such a way that during its operative lifetime it has to calculate the minimum amount of hashes.

The subgroup of links of the hash chain to be stored will divide the chain in equal sub-chains, with the last sub-chain (the one that ends with the top hash) will be divided by its half, and the second part divided by its half, and so on. The number of hashes (links of the chain) that will be stored with its correspondent position depend on two things: the memory available for storage and the maximum number of hashes that a node must calculate to obtain the last not used link of the chain (which corresponds to the length between the seed and the first stored hash link).

Each time a sensor nodes issues a data message, it precedes the data that it has sensed by a node identifier, this node identifier can be the top hash since (when long enough) is going to be statistically unique.

After the node identifier and the data, it will include a Message Authentication Code (MAC) of the information using the link n of the chain and revealing the link $n+1$ of the chain. Therefore, each node that receives it will receive the current information and be able to authenticate the information sent in the previous message.

The hash h_n of the hash chain is coded with two fields: the number of the link in the chain and the hash value. Therefore, if some messages are lost, the sink can still perform delayed verification because it knows how many times the new hash value has to be hashed.

C. Considerations

About how to merge the security requirements and aggregation at intermediary nodes, there are several considerations:

1) An aggregator node only will start to aggregate the received data after it has received the second transmission of an specific sensor. The first message of this specific sensor received by the aggregator is just forwarded to the sink without processing it (the aggregator only records the required hash-related fields in order to authenticate the next message received from that sensor). The

second message received from the specific sensor will be aggregated as it can be authenticated. However, in order to authenticate the previous message by the sink, which requires the hash-based fields from this second message, the aggregator builds an extra packet with these fields and sends it to the sink (these fields could be concatenated to the same packet in which the data is aggregated).

2) Aggregators use its own hashes to authenticate the aggregated data. Probably, higher security requirements should be considered for those nodes.
3) Notice that this mechanism is only useful if the aggregation process or data fusion (e.g. to compute the average of a set of values, to concatenate measures without process them, etc.) allows it.
4) Aggregators only aggregate messages which have not been aggregated before. It means that packets which have been already aggregated will be only forwarded by the other aggregators in the path to the sink.

IV. Performance results

To evaluate the protocol previously described a network simulator has been developed using the COST (Component Oriented Simulation Toolkit) package [15]. A network formed by N sensor nodes distributed over a plain area has been simulated, each node generates λ packets of a fixed length L_{data} per second on average. It has been considered that the resulting fusion of all data received in the interval could be reduced to fit into a single packet of length L_{data}. The benefits of using aggregation in the intermediary nodes have been compared with the case in which all packets are forwarded to the sink. Moreover, the secure solution proposed has been compared with the non-secure case, the main difference between them is the length of data packets, in the secure solution the packet length is composed of five different fields (due to the use of delayed authentication with hash chains):

- Sensor identifier (which can be the TOP-HASH) of 128 bits, which is the length of the $MD5$ hash.
- The Message Authentication Code of the data (128 bits long).
- Identifier of the link in the hash chain (4 bits).
- The hash link corresponding to the previous data (128 bits long).
- Payload $L_{data} = 10$ Bytes.

While in the non-secure case, the application packet contains only the Data field. The network layer (6 Bytes) and the Link Layer and Medium Access Control (24 Bytes) headers are added to all the data packets transmitted.

The channel capacity is assumed to be constant and equal to $C = 10\ Kbps$ over the wireless link and ideal channel conditions have been considered. Each sensor node generates and sends data at a rate equal to $\lambda = 1/10\ packets/s$. When a node is configured to aggregate packets, it waits for a certain amount of time $D_{agg} = 2.5$ s (on average two packets are aggregated) and sends only one packet containing the result obtained from the fusion function applied to the data received. On average, intermediary nodes are responsible of aggregating data of 8 sensors.

The results obtained from the simulations are shown in Figures 3, 4 and 5 with 95% confidence intervals. In Figure 3 the total traffic sent by the aggregation nodes is plotted. It could be seen that by using security the total load of the network increases but it could be reduced by using aggregation techniques. In this particular case, the load of using security and aggregation is similar to the load obtained without security and without aggregation.

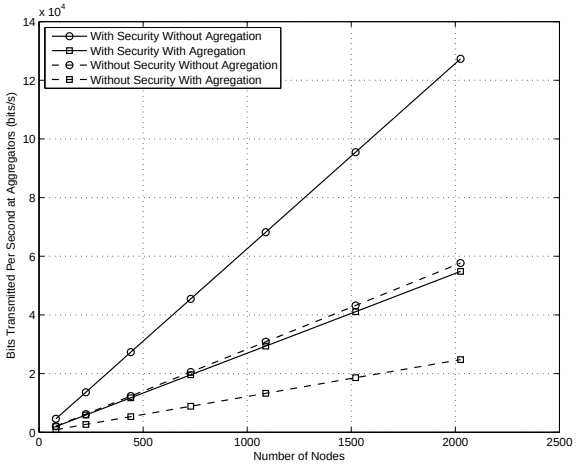


Fig. 3. Impact of security and aggregation on the information transmitted

Figure 4 shows the total traffic received (transmitted - lost) at sink. The saturation point of the network could be derived from this figure, notice that without aggregation the network saturates with approximately 400 nodes (with security) and 700 nodes (without security), but using aggregation the number of nodes supported increases to 700 using security and 1500 without security. Higher aggregation delays would allow to obtain a higher number of nodes in unsaturated conditions.

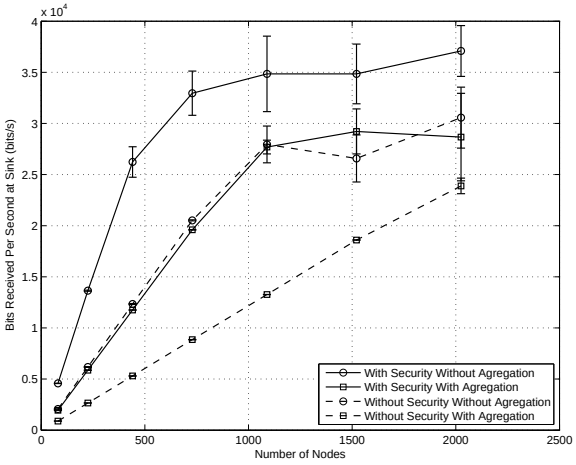


Fig. 4. Impact of security and aggregation on the information received

Figure 5 plots the average end-to-end delay for the scheme proposed with and without secure functions and aggregation. Notice how the delay increases noticeably when the saturation point is reached. The prize to be paid by using aggregation is an extra delay caused by the aggregation delay (D_{agg}). It is important to see that this delay is only higher when the saturation point is not reached, when saturation appears the aggregation delay is smaller than the delay obtained without using aggregation for the same number of nodes.

When adding security, the network load increases in both cases leading to a reduction in the network size. Therefore, the cost of using security has to be considered in terms of a reduction of the network size and/or the frequency of data transmission by the sensors. Aggregation could be used in order to reduce the load of the network and then reduce the packet delay.

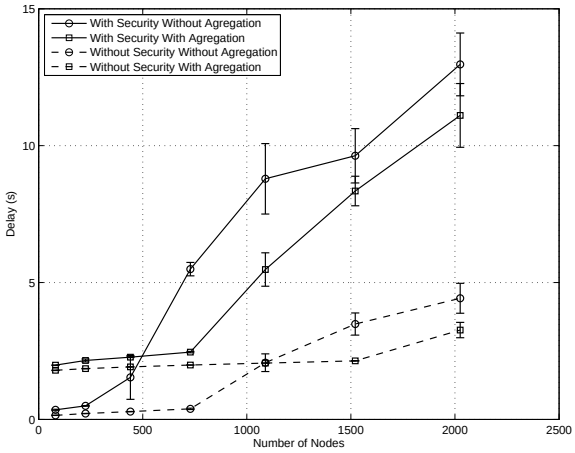


Fig. 5. Impact of security and aggregation on the packet delay

V. Concluding Remarks

This work introduces a set of preliminary ideas to be applied in securing low-capability sensor networks. It shows an efficient way to perform data collection in sensor networks while providing different security mechanisms to secure data transmission depending on the needs of the sensor network and the capability of the nodes.

The network can be secured by using shared secrets between the sink and the sensor nodes that might be shared before or after network deployment or, when shared secrets between sink and sensor nodes before deployment is a too strong requirement and sensor nodes do not have the computational capacity to calculate signatures, by the use of hash chains.

Network efficiency (less battery consumption, higher network size, etc.) is improved by using an asynchronous data aggregation mechanism which is shown as valid mechanism for non-real time sensor networks as it reduces the extra bits due to unnecessary overheads.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. IEEE Commun. Mag. 40 (8) (2002) 102–114., 2002.

[2] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad hoc on-demand distance vector (AODV) routing. Internet Request for Comments RFC 3561, November 2003.

[3] T. Clausen, P. Jacquet (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot. Optimized link state routing protocol (olsr). RFC 3626, October 2003. Network Working Group.

[4] R. Hauser, A. Przygienda, and G. Tsudik. Reducing the cost of security in link state routing. In *Symposium on Network and Distributed Systems Security (NDSS '97)*, pages 93–99, San Diego, California, February 1997. Internet Society.

[5] Steven Cheung. An efficient message authentication scheme for link state routing. In *13th Annual Computer Security Applications Conference*, pages 90–98, 1997.

[6] Adrian Perrig, Robert Szewczyk, Victor Wen, David E. Culler, and J. D. Tygar. SPINS: security protocols for sensor netowrks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 189–199, 2001.

[7] Kan Zhang. Efficient protocols for signing routing messages. In *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS'98)*, July 2001.

[8] N. Asokan. Presentation at an informal workshop on mobile and ad hoc networking security, EPFL, Lausanne, December 2001, December 2001.

[9] Manel Guerrero Zapata and N. Asokan. Securing Ad hoc Routing Protocols. In *Proceedings of the 2002 ACM Workshop on Wireless Security (WiSe 2002)*, pages 1–10, September 2002.

[10] Yih Chun Hu, Dave Johnson, and Adrian Perrig. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02), June 2002*, pages 3–13, June 2002.

[11] Josh Broch, David A. Maltz, David B. Johnson, Yih Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th Annual International Conference on Mobile Computing and Networking*, pages 85–97, 1998.

[12] Bhaskar Krishnamachari, Deborah Estrin, and Stephen B. Wicker. The impact of data aggregation in wireless sensor networks. In *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 575–578, Washington, DC, USA, 2002. IEEE Computer Society.

[13] Konstantinos Kalpakis, Koustuv Dasgupta, and Parag Namjoshi. Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. *Comput. Networks*, 42(6):697–716, 2003.

[14] Bartosz Przydatek, Dawn Song, and Adrian Perrig. SIA: Secure information aggregation in sensor networks. In *ACM SenSys 2003*, Nov 2003.

[15] Gilbert Chen and Boleslaw K. Szymanski. Cost: A component-oriented discrete event simulator. In *Proceedings of the 2002 Winter Simulation Conference*, pages 776–782, 2002.