n-LQA: n-Layers Query Authentication in Sensor Networks

Ioana Rodhe, Christian Rohner and Andreas Achtzehn
Department of Information Technology, Uppsala University
Email: {ioana.ungurean, christian.rohner}@it.uu.se, andreas.achtzehn@rwth-aachen.de

Abstract

We present a protocol for query authentication in a sensor network where there is multi-hop communication and the queries are broadcasted by the base station into the network. Authenticating the queries is important so attackers cannot modify existing queries because this would lead to wrong readings; or insert new ones into the network because this would lead to waste of energy. We propose a layered query authentication protocol that ensures that, in the presence of less than n captured nodes, unauthorized queries are stopped after a small number of hops. When more than n nodes are captured, the unauthorized queries will only spread in one direction with a limited angle. Message authentication codes (MACs) are used to protect the authenticity and integrity of the query. n MACs are attached to the query message at the base station and the nodes replace MACs from this message in an interleaved manner.

1. Introduction

Sensor networks consist of many small devices that are used to sense the environment. The sensors are usually cheap, small devices with battery and memory constraints and little computation power. When the sensors are deployed on a large surface, multi-hop communication is used because of their short transmission range.

A special node, typically called base station, is used to query the nodes for sensor readings. Most query dissemination methods, for example TinyDB [6] or Directed Diffusion [5], are variants of query flooding with differences in the ways they direct a query towards a specific direction. The base station is assumed to be a powerful and tamper-proof device.

A sensor node can have many sensing devices, e.g. movement, temperature, light sensors and it might have possibility to locally save the readings for some time. The queries can specify the location of the sensor nodes that should send their readings, which sensor readings that should be sent or from which time interval. If the query is modified, the sensor will answer with different readings than required. The base station cannot verify if the nodes answer to the original or to a modified query because the answers include only data and no informatin about the content of the data. So it is very important that the nodes are able to authenticate the query as coming from the base station.

Another important aspect is protecting against unauthorized queries being inserted into the network because disseminating a query into a sensor network causes an action at all sensor nodes that receive the query and thus is a resource critical operation. These aspects are of particular importance as the nodes might be placed in unattended and possibly hard-to-get-to places where it is hard to replace batteries.

The goal of this work is that every node in the network, regardless of how far it is from the base station, can authenticate a query. The solution we propose makes use of a layered network with layer-specific secret keys which are used to achieve interleaved authentication of the queries. The protocol, called n-*LQA*, ensures that in the presence of less than n captured nodes, unauthorized queries are stopped after a small number of hops which is at the most the number of captured nodes. Even when capturing more than n nodes, the unauthorized queries will only spread in one direction with a limited angle. The attacker, to be sucessful, has to capture nodes from consecutive layers. We also use pairwise keys so that, at each hop, the receiving node can verify the identity of the sending node in order to protect against node impersonation.

The rest of the paper is organized as follows. In Section 2 we discuss related work, then we present the n-LQA protocol in Section 3 together with the network settings and attacker model. An evaluation of the protocol together with simulation results are presented in Section 4 and we conclude the paper in Section 5.

2. Related work

F. Armknecht et al. [1] present a protocol for query authentication in which all the nodes share a common key with

1-4244-1455-5/07/$25.00

the base station and from this key a hash chain is generated. The values from the hash chain are used to sign queries sent by the base station so the nodes can authenticate them. The protocol, due to use of hash chains, requires that every node in the network calculates the currently used key from the chain, which is an energy consuming task, and that the queries are sent to the whole network, so the nodes can keep track of which key from the chain is used. The latter is not desired because many queries might be meant for just a part of the network. They also assume that the sensor nodes are tamper-proof so, when captured, the secret key they share with the base station cannot be retrieved. If the sensors would not be tamper-proof, capturing one node would reveal the secret key and fake queries could be accepted by the network.

Perrig et al. [7] proposed μTESLA, a protocol that provides authenticated broadcast for sensor networks. The protocol uses key chains and requires that the base station and the nodes are loosely time synchronized. The protocol achieves asymmetry by a delayed disclosure of the symmetric keys. However, time synchronization in large sensor networks is hard to achieve.

Benenson et al. [2] proposed a probabilistic query authentication protocol that uses 1-bit message authentication codes (MACs). In this protocol each sensor node is preloaded with keys chosen randomly from a large key pool and, for each query, a number of 1-bit MACs are computed using keys chosen from the same key pool. When receiving a query, the sensor node has, with some probability, some of the keys used to calculate the 1-bit MACs and can verify the authenticity of the query. To increase the chances of discovering a fake query, the number of 1-bit MACs has to be large, resulting in increased message length.

3. n-LQA protocol

We present the deterministic n-layers query authentication protocol (n-LQA) that allows for a limited number of node capture and does not require synchronization.

3.1. Network settings and attacker model

We assume query flooding and organize the sensors in layers: nodes i hops away from the base station comprise layer i. When a node in layer i broadcasts a query message, only the neighbor nodes in layer $i+1$ will deal with the query. We consider that we have m layers in the sensor network and we refer to a node as u^i, v^i or w^i where i denotes the layer. In Figure 1 we show a sensor network with $m = 5$ layers.

We assume that in the deployment phase a wave algorithm, starting from the base station, is used to determine the layers in the network and that nodes exchange layer information with their neighbors. We also assume that the nodes will remain in the same layer during the lifetime of the network.

The attacker is interested in modifying the existing queries and inserting new unauthorized queries into the network. We assume that attackers can capture nodes and we do not consider nodes as being tamper-proof. Once a node is captured, the attacker will be able to read its memory and find out all its keys. He will also be able to reprogram the nodes to work in his favor and to copy keys from one node to another. However, we assume that attackers cannot capture a very large number of sensor nodes without being detected.

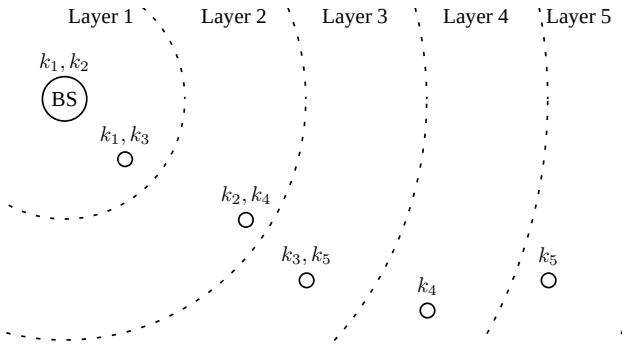


Figure 1. Layers in a network with one node in each layer. k_i denotes the *authentication key* of layer i. BS is the base station and $\circ$ are the sensor nodes.

3.2. Cryptographic primitives and keys

The nodes in layer i share a common secret key k_i, called *authentication key*, from which one-time secret keys $k_{i,r} = E(k_i, r)$ are derived using a one-way function E and a random number r. The random number r is generated by the base station for every query and sent together with the query so that the nodes can compute $k_{i,r}$. The authentication key k_i is also known by the base station when $i \leq n$ or by all nodes in layer $i - n$, when $n < i \leq m$. n is a security parameter chosen based on the security requirement of the sensor network. An example of key distribution is shown in Figure 1 where we consider $n = 2$. One way to deploy these keys into the nodes is to let the base station generate and send them to each node. The base station uses individual keys, shared with each node, to securely send the authentication keys. Individual keys are often used for secure communication between the base station and individual nodes and they are usually preloaded

into the nodes before deployment.

Message authentication codes (MACs) are used to protect the authenticity and integrity of the queries. These MACs are computed using the one-time secret keys, $MAC(k_{i,r}, q)$, where $i = 1 \ldots m$ and q is the query. We say that a node accepts a query if it can check one of the MACs that are sent together with the query.

3.3. Protocol description

Base station: When the base station sends a new query q into the network, it generates a random number r and uses it to compute one-time secret keys $k_{i,r}$ for the first n layers. The one-time secret keys are used together with the query to compute the MACs. The query, together with the base station's identity, the random number r, and the n MACs is then sent into the network.

Each sensor node: When receiving a new query, a node in layer i calculates the one-time secret key $k_{i,r}$ using the random number r included in the query and verifies the $MAC(k_{i,r}, q)$. If the query is authentic, it removes $MAC(k_{i,r}, q)$ from the message and adds $MAC(k_{i+n,r}, q)$ (if layer $i + n$ exists in the network), thus interleaving the authentication process.

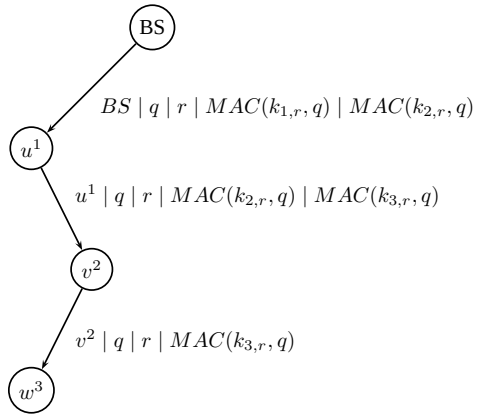


Figure 2. Nodes in a sensor network where $n = 2$ and the query message that is sent into the network.

Example: In Figure 2 we consider a sensor network with $m = 3$ layers and $n = 2$ and show how the query message is sent to nodes u^1, v^2 and w^3. The message sent by the base station BS includes the identity of the sending node (the base station), the query q, the random number r and MACs calculated with the keys $k_{1,r}$ and $k_{2,r}$,

$$BS \rightarrow u^1 : BS \mid q \mid r \mid MAC(k_{1,r}, q) \mid MAC(k_{2,r}, q).$$

The node u^1 will check $MAC(k_{1,r}, q)$ and calculate $MAC(k_{3,r}, q)$. The message sent by u^1 will include q, r, the MAC that the base station calculated with the key $k_{2,r}$ and the MAC that u^1 calculates with the key $k_{3,r}$,

$$u^1 \rightarrow v^2 : u^1 \mid q \mid r \mid MAC(k_{2,r}, q) \mid MAC(k_{3,r}, q).$$

Node v^2 checks $MAC(k_{2,r}, q)$ and does not have to calculate any MAC because $n = 2$ and $m = 3$,

$$v^2 \rightarrow w^3 : v^2 \mid q \mid r \mid MAC(k_{3,r}, q).$$

These messages are broadcasted, so other nodes from these layers will receive them. We showed only these particular nodes for the ease of explanation.

3.4. Node impersonation

When receiving a query, a node (in layer i) first checks where the query comes from, because as mentioned in Section 3.1, the only messages that the node is interested in are the ones that come from the nodes in layer $i - 1$. In our protocol, the query message includes the identity of the node that is sending the message (which can be a node id or name). The problem with this is that nodes can easily impersonate other nodes, they only have to find out their identity. A captured node from layer i can impersonate a node from layer $i - 1$ in order to make his neighbors from layer i accept a modified or inserted query. The nodes that accept the query will forward it to nodes in layer $i + 1$ which will drop the query as long as the attacker does not have k_{i+1}. In Figure 3 we show how long a message will be forwarded into the network. Consider that node v^i is captured and that it modifies a query q to q'. By impersonating node u^{i-1} it is able to convince node w^i to accept the modified query q'. If node v^i does not know k_{i+1} it cannot recalculate $MAC(k_{i+1,r}, q)$ for the modified query q' and nodes x^{i+1} and y^{i+1} will not accept the query as being authentic. But if node v^i knows k_{i+1}, then nodes the x^{i+1} and y^{i+1} will accept the message as being authentic and forward it. w^i broadcasts the message so it is only one message that is sent to both x^{i+1} and y^{i+1}. As sending messages is an expensive operation it is not desired that these modified or inserted queries are forwarded.

3.5. Pairwise keys

We propose the use of pairwise shared keys between neighboring nodes that are in different layers to protect against node impersonation. We refer to these keys as $k_{u,v}$, where u and v are neighbors. These keys are established once in the deployment phase. Some mechanisms to establish pairwise keys can be found in [3] and [9].

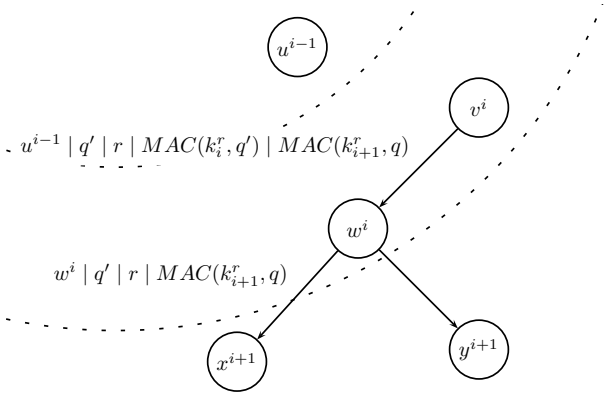


Figure 3. Node v^i has been captured and impersonates node u^{i-1} to make node w^i accept the modified query q'.

If, as said in Section 3.2, each node knows in which layers its neighbors are, then when forwarding a query message, MACs calculated with the pairwise keys that it shares with the nodes in the next layer are also included in the message. We call these *pairwise MACs*.

When a node receives a query message it will first check the pairwise MAC to verify the identity of the sender and then check the MAC calculated with the authentication key. This way nodes cannot impersonate other nodes unless they know the pairwise keys of the respective nodes.

4. Evaluation

Our protocol protects against query modification or insertion when intruders can capture up to $n-1$ nodes. In this Section we discuss what happens when nodes are captured, depending on the number of captured nodes and their position in the network, and describe how n should be chosen. We assume the use of pairwise keys.

4.1. Modifying queries

If an attacker manages to capture node u^i in layer i, he gets access to at most two keys, k_i and k_{i+n} (if there is a layer $i+n$). With these two keys he would be able to convince all the nodes in layers i and $i+n$ to accept a modified or inserted query q'. However, the query has to reach these nodes. When q' sent by u^i reaches the nodes in layer $i+1$ the query will be rejected because the attacker does not know the secret key k_{i+1} so he will not be able to generate a valid MAC.

If an attacker captures $s < n$ nodes from s consecutive layers, starting from layer i, only nodes in these layers will accept the query. When the query reaches the nodes in layer $i+s$ it will be dropped because verification of the MAC will fail.

Only if an attacker captures $s \geq n$ nodes from s consecutive layers, starting from layer i, he would be able to convince nodes in the layers j, where $i < j \leq m$, to accept the query. The reason is that he has enough keys (at least n) to generate valid MACs that these nodes will verify as coming from the base station. These nodes are the nodes that can be reached when the messages are sent from the corrupted nodes. The nodes in the layers above layer i will not accept the query, because the attacker cannot get access to their authentication keys.

A modified query will only spread in a limited angle within the network. Figure 4 shows the nodes that are reached by an unauthorized query in a sensor network where an attacker has captured at least n nodes from consecutive layers starting with the first layer. The query is sent from a captured node from the first layer and, as mentioned, only its neighbors from the second layer are interested in it and will forward it. In turn, only their neighbors from the third layer are interested in the message and so on. As we see from the figure, the reached nodes form a tree-like structure spreading from the base station towards the outer part of the network in a limited angle. The angle depends on the captured node's position within the layer (the closer to the outer border of the layer, the bigger the angle will be). So, even if an attacker captures nodes from the outer layers, where there are many nodes in each layer, the modified query will only spread in a limited part of these layers .

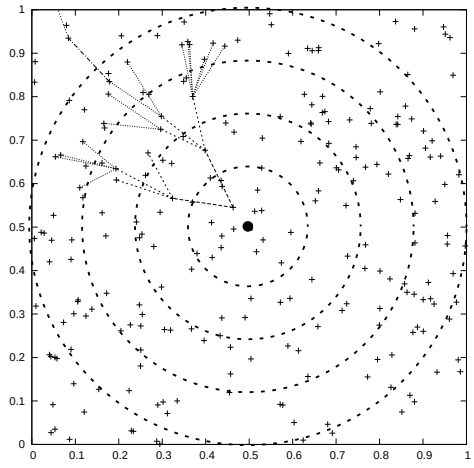


Figure 4. The nodes that are reached by an unauthorized query when $s \geq n$ nodes are captured. The base station is placed in the middle of the area. The dashed circles approximately indicate the layers.

4.2. Choosing n

An important aspect of the protocol is choosing n, which is the number of MACs that are sent together with a query in addition to the MACs computed with the pairwise keys. n should not be smaller than two since that would disable the interleaved functionality of the protocol, and capturing an arbitrary node would allow to modify or insert queries from that node. However, the spreading of the message will still follow the pattern in Figure 4. The upper limit for n is naturally the number of layers in the network. In this case the base station knows all the authentication keys and it will send the query with all the MACs. Setting n to the number of layers provides the best possible security for this protocol because an attacker needs to capture one node from every layer if he wants to modify or insert messages.

The message length grows linearly with n, because n MACs are sent together with the query. Hence, a bigger n requires more power consumption in the nodes when transmitting and receiving query messages. Therefore, choosing n is a tradeoff between the achieved security level and power consumption. However, every node generates and verifies only one MAC per query irrespective of the size of n.

4.3. Simulation results

We used Matlab to simulate our protocol. As a network parameter we define *node density* d to be the average number of neighbors of a node and in our simulations we use $d \in \{7, 12\}$. We uniformly and randomly distribute 250 nodes in an area and adjust their transmission range to achieve the desired node density d. The base station is positioned in the middle of the area.

d is an important parameter of the sensor network because it influences the network capacity. The optimal value for d is discussed in [4], where the authors suggest 6 and 8. Xue et. al. [8] shows that for a network with N nodes, the node density should be at least $5.1774 \log N$ to ensure overall connectivity, which results in $d = 12$ for our simulations.

In each simulation run, we generate a new network and use c captured nodes that are chosen randomly from the nodes in the network and from these c nodes we choose at random one to start sending the unauthorized query. We consider that the keys retrieved from all the captured nodes are copied into the node that will start sending the query. We are interested in the number of nodes that accept and forward the unauthorized query given c and different n. We refer to these nodes as *reached nodes*. We run the simulation 500 times for each combination of parameters.

Figure 5 shows the percentage of reached nodes in the network for $c = 2$, 4 and 8 captured nodes and $n = 1 \ldots 12$ MACs sent together with the query. In Figure 5(a) the node density is 7 and in Figure 5(b) the node density is 12.

When sensor networks are generated, the number of layers varies. Therefore we choose n from 1 to the mean number of layers from 100 simulations with $d = 7$. We include the case $n = 1$ as reference to show the strength of the interleaved functionality of the protocol, as discussed in Section 4.2. Figure 5 shows that much more nodes are reached when $n = 1$ than when $n \geq 2$. Because the spreading of a query follows the pattern in Figure 4, even when $n = 1$ the number of reached nodes is still limited to one part of the network.

From Figure 5 we can also observe that the number of reached nodes does not differ very much for $n \geq 4$. Because of this we can choose a small n resulting in a smaller message length without compromising the security of the network. For example when $d = 7$, n can be chosen 4 or 5.

When simulating the protocol, the c captured nodes are chosen randomly. Thus they do not have to be from different nor consecutive layers. So, the number of consecutive layers s the attacker has keys from might be smaller than n even when $c \geq n$. Because of this we cannot see a drop of reached nodes when $n > c$ as one might expect.

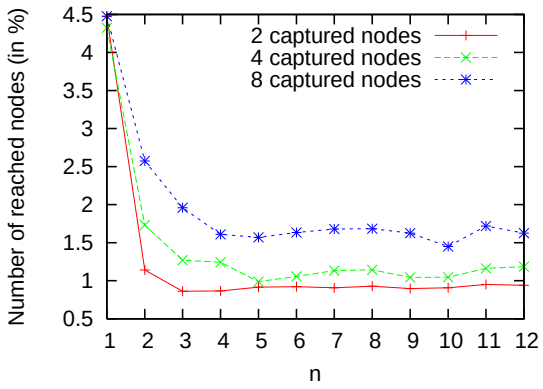
4.4. Pairwise keys

We have also simulated the worst case scenario where an attacker captures nodes from at least n consecutive layers starting from layer 1 and modifies existing queries or inserts new unauthorized ones into the network by sending them from the captured node in layer 1. We are interested in the number of nodes that receive this query. Because the attacker captured enough nodes to be able to compute MACs for the new query, all the nodes that receive the query will consider that it is sent by the base station. We have considered two scenarios: the first one when pairwise keys are not used and the attacker uses the node captured in layer 1, say u^1, to impersonate the base station and like this convincing u^1's neighbors from the same layer to accept the message too; and the second case where pairwise keys are used, so the attacker cannot impersonate nodes. In Table 1 are the results of the simulation.

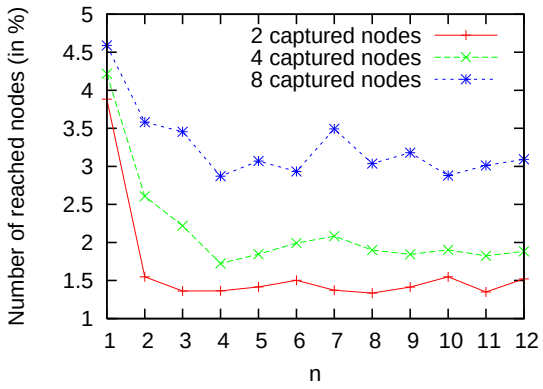
As we can see from Table 1 using pairwise keys considerably decreases the number of nodes that receive the modified or inserted query.

5. Conclusions

We proposed a deterministic n-layered query authentication protocol which ensures that in the presence of less that n captured nodes, unauthorized queries are stopped after a small number of hops. Even when the attacker captures more than n nodes, the unauthorized query will only



(a) $d = 7$



(b) $d = 12$

Figure 5. Number of reached nodes (in %) when $c = 2, 4, 8$ captured nodes

spread in one direction. Moreover, the attacker needs keys from consecutive layers, so he might have to capture more than n nodes. The value of n influences the length of the query messages, and thus also the power consumption in each node when transmitting and receiving, so n should be chosen as small as possible. The simulation results indicate that $n = 4$ is sufficient for most scenarios and that unauthorized queries spread only in a small part of the network.

For future work we plan to investigate support for nodes to switch layers because of changes in network connectivity.

References

[1] F. Armknecht, J. Girao, M. Stoecklin, and D. Westhoff. Re-visited: Denial of service resilient access control for wireless sensor networks. In *Third European Workshop on Security and Privacy in Ad hoc and Sensor Networks*, Hamburg, Germany, September 2006. ESAS2006. Held in conjunction with ESORICS 2006.

	Nodes reached (in %)	
	without pairwise keys	with pairwise keys
d = 7	55%	28%
d = 12	63%	24%

Table 1. The number of nodes (in %) that receive a modified or inserted query and consider it authentic, for two different node densities

[2] Z. Benenson, L. Pimenidis, F. C. Freiling, and S. Lucks. Authenticated query flooding in sensor networks. In *PERCOMW '06: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, page 644, Washington, DC, USA, 2006. IEEE Computer Society.

[3] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proc. of the IEEE Security and Privacy Symposim 2003*, 2003.

[4] T. Hou and V. Li. Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications*, 34(1):38–44, 1986.

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1), 2003.

[6] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 491–502, New York, NY, USA, 2003. ACM Press.

[7] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar. SPINS: security protocols for sensor netowrks. In *Mobile Computing and Networking*, 2001.

[8] F. Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wirel. Netw.*, 10(2):169–181, 2004.

[9] S. Zhu, S. Setia, and S. Jajodia. Leap: efficient security mechanisms for large-scale distributed sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 62–72, New York, NY, USA, 2003. ACM Press.