

Re-visited: Denial of Service Resilient Access Control for Wireless Sensor Networks

Frederik Armknecht¹, Joao Girao¹, Marc Stoecklin², and Dirk Westhoff^{1*}

¹ NEC Europe Ltd.

{frederik.armknecht,joao.girao,dirk.westhoff}@netlab.nec.de

² Swiss Federal Institute of Technology - Lausanne

marc.stoecklin@epfl.ch

Abstract. The appliance of wireless sensor networks to a broad variety of applications doubtlessly requires end-user acceptance. End-users from various computer network unrelated disciplines like for example from the agriculture sector, geography, health care, or biology will only use wireless sensor networks to support their daily work if the overall benefit beats the overhead when getting in touch with this new paradigm. This does first and foremost mean that, once the WSN is deployed, it is easy to collect data also for a technical unexperienced audience. However, the trust in the system's confidentiality and its reliability should not be underestimated. Since for end-users from various disciplines the monitored data are of highest value they will only apply WSN technology to their professional activities if a proper and safe access control mechanism to the WSN is ensured. For FIPS 140-02 level 2 or level 3 conform sensor devices we provide an access control protocol for end-users of civilian WSN applications that i) ensures access to the monitored data only for authorised parties, ii) supports user-friendly data queries and iii) is DoS resilient to save the sensor nodes' battery capacity.

1 Introduction

Recently considerable contributions have been made in the area of wireless sensor networks (WSN) to effectively request and receive environmental data from a WSN. We observe two principle directions to apply a WSN: The first type of WSNs, which we term *synchronous* WSNs are WSNs where the monitored data is fluctual and is most likely to be used for some real-time control monitoring. Data are transmitted in a push or in a pull mode. In contrast, we define an *asynchronous* WSN as one that provides information to an authorised reader only seldomly. Here the data provision to the end-user is exclusively in pull mode.

* The work presented in this paper was supported by the European Commission within the STReP UbiSec&Sens of the EU Framework Program 6 for Research and Development (IST-2004-2.4.3). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the UbiSec&Sens project (<http://www.ist-ubisecsens.org>) or the European Commission.

The network continuously monitors and stores environmental data as a function over the time and/or over the monitored region. Consequently, after a period of monitoring and storing, the WSN contains a very fine granular environmental fingerprint. Such information can be requested via some external reader device by the end-user, e.g. a winemaker, a geologist, or other professionals.

For synchronous WSNs Madden et al. in [10] and Hellerstein et al. in [7] provide an SQL-based query model for tiny in-network aggregation in WSNs addressing specific monitoring durations of the network. Queries address monitoring periods in the present and in the future. However, although in [7] the concept of storage points allows to buffer a streaming view of recent data, the fully-fledged architecture to store monitored data of an event of the past within the WSN is not addressed in their work.

With respect to asynchronous WSNs the problem of how to use the limited persistent storage capacity of an asynchronous WSN to store sampled data effectively has been discussed by Tilak, Abu-Ghazaleh and Heinzlmann in [13]. The authors provide a cluster-based collaborative storage approach and compare it to a local buffering technique. Collaborative storage is a promising approach for storage management because it enables the use of spatial data aggregation and redundancy control among neighboring sensors to compress the stored data and to optimize the storage use.

Although it is unrealistic to expect that all the aforementioned approaches survive the competition to market, such a diversity of storage architectures and information gathering concepts for WSNs on the one side, and the ultimate need for a simple, user-friendly and consistent query interface on the other side, manifests that a flexible data access framework will be a coactive part of future service-oriented WSNs. Moreover, since for end-users from various disciplines the collected data are of highest value they will only apply WSN technology to their professional activities if a proper and safe data management is ensured. This includes the encrypted storage and transport of data which have been monitored over the time but also mechanisms to access the collected data in an authenticated way. For the first we proposed solutions in [15] for synchronous WSNs and in [6] for asynchronous WSNs. It is the contribution of this work to provide a secure and efficient access control mechanism to allow only an authenticated reader to request data from the WSN. We assume a distributed and encrypted storage- and transmission architecture for reliable long-term storage of data in the WSN as well as a translation framework to map user-friendly database queries into controlled flooding messages. For the latter we introduce a generic query translation model which we subsequently apply to show that the proposed access control protocol for WSNs is applicable to any WSN database architecture. An early version of this work has been presented at [16].

When considering access control, one must define the assets one is protecting and the environment. In this particular model, we protect access to the network and, in doing so, we provide an authorization mechanism which only allows a valid reader to perform one query per interaction with a supervising entity, the sink. In parallel we provide a mechanism which fits the query model described

in Section 3 and provide the link between the user world and the WSN world and bind it to the access control information which can be validated at both sink and sensors.

2 Network Model and Device Characteristics

The WSN considered in this work is static and densely distributed. It is presented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ with $|\mathcal{N}|$ nodes and $|\mathcal{L}|$ links. Each node represents a wireless sensor node, e.g., a MicaZ mote, and each link represents a bidirectional communication channel over a shared medium, e.g., the RF channel specified by IEEE 802.15.4 WPAN. There is one single stated node S , the sink node. The sensor nodes that compose a WSN are typically small in size, wireless, and have very limited communication, computation, storage and power capabilities. For example, the Berkeley Sensor Motes use an 8-bit 4MHz microcontroller with 4KB of memory and a radio transceiver with a maximum 10 kbps data rate. One consequence of limited computing and storage capacities is that modular arithmetic with large numbers is difficult and, therefore, asymmetric cryptography should only be used very carefully. In particular, standard Diffie-Hellman key exchange protocols and even low exponent RSA techniques are prohibitively expensive for sensors. Based on the above, extremely low-cost mechanisms are needed. Furthermore, it has been pointed out that, in WSNs, sending a bit is roughly 10^2 times more expensive than executing a processor instruction [8].

The NIST standard FIPS 140-02 [4] defines four levels of physical security for cryptographic devices. In this work we assume the sensor nodes being level 2 or level 3 devices, namely devices that implement tamper evidence mechanisms or devices that in addition to tamper evidence mechanisms also implement tamper response mechanisms like top-metal sensor meshes or light-sensors. An adversary, who destroys a sensor line or shortens it to ground or power, causes the device to self-destruct. Partly protected devices prevent clever outsiders to read out sensitive information from the sensor nodes. However, knowledgeable insiders or funded organisations which spend several 100,000 EURs for an attack can clearly read out the data. We expect FIPS 140-02 level 2 and even level 3 enabled devices to be reasonable in future in terms of costs also for the usage for some types of WSN.

3 Query Translation Model

The goal of this paper is to present a denial of service resilient access control for WSN applications that support "user-friendly" data queries. However, we will see that the "user-friendliness" imposes conditions on the communication between reader and sensor nodes that have to be considered in an access control protocol.

For a "user-friendly" distributed database, it is mandatory that any query $q \in \mathcal{Q}$ from the end-user's *query space* can be mapped to a "network-friendly" controlled *flooding message* $\varpi \in \mathcal{M}$, i.e., the existence of a function $f : \mathcal{Q} \rightarrow \mathcal{M}$.

However, in practice f should provide a semantical appropriate interpretation of q . This means that ϖ should reflect the user’s query as good as possible on the WSN’s side. We will call such a function *adequate*.³ But here the following problem arises. On the one hand, ϖ relies on the topology of the WSN. Therefore we cannot expect that an “end-user friendly” query from the user space can *adequately* be translated into a query from the WSN query message flooding space unless additional information are available. On the other hand, the user should be bothered only to a minimum with the technical details of the WSN. The queries should contain as much information as possible but not more than necessary.

Let the domain \mathcal{T} represents the topology information of the WSN. Obviously, it is only available after the roll-out of the network. Some information on \mathcal{T} are certainly necessary to formulate a “network-friendly” flooding message. On the other hand, major parts of \mathcal{T} should be hidden from the end-user as otherwise “user-friendliness” would not be fulfilled. We therefore divide \mathcal{T} in $(\mathcal{T}_U, \mathcal{T}_N)$ whereas \mathcal{T}_U represents only those information about the WSN’s which are relevant for the end-user to generate context-sensitive user queries. Contrary, \mathcal{T}_N contains topology information which are mandatory to formulate WSN architecture specific flooding messages and which should be hidden from the end-user.

Consequently, to get an adequate function, we consider functions $f^* : \mathcal{Q} \times \mathcal{T} \rightarrow \mathcal{M}$ instead of $f : \mathcal{Q} \rightarrow \mathcal{M}$. Given such a mapping f^* , we can doubtlessly infer that such a translation exists and we can build a framework that can translate (q, t_U) with t_U from \mathcal{T}_U into an ϖ for a specific WSN architecture. Examples for WSN architectures are *tinyDB*, *TAG* or *tinyPEDS*.

Remarks

- With respect to the entropy of a message in principle it holds $|q| = |\varpi|$. Although due to the involvement of \mathcal{T} in f^* one could expect $|\varpi| > |q|$. We argue that the gain of information comes with the implicit knowledge about the network’s topology and it does not need to be transmitted in the message itself.
- the domain \mathcal{M} relates to the WSN’s architecture. Known WSN architectures are e.g. *tinyDB*, *TAG*, *tinyPEDS*, etc. Any $\varpi \in \mathcal{M}$ considers the architectural semantic and thus we point out that for different WSN architectures the domain \mathcal{M} is a separate one.
- the domain \mathcal{T} relates to the WSN topology. In principle $t = (t_U, t_N) \in \mathcal{T}$ is dynamic. For simplicity we assume t to be static. It describes the topology of the WSN directly after the roll-out.
- The existence of an adequate mapping f^* can best be proven over the detour of comparing $r_U \in \mathcal{R}_U$ with $r_N \in \mathcal{R}_N$ with $f_U : \mathcal{Q} \rightarrow \mathcal{R}_U$ and $f_N : \mathcal{M} \rightarrow \mathcal{R}_N$: $r_U \stackrel{?}{\approx} r_N$ (approximation to the expected value). \mathcal{R}_U and \mathcal{R}_N represent

³ An example for a non-adequate function f would be one that maps any query q to the same flooding message ϖ . This is principally possible but wouldn’t make any sense in practice.

the query response space from the user’s perspective respectively from the network side.

- The topology \mathcal{T}_N can automatically be generated by the WSN initially after the roll-out of the WSN. However, the mapping to \mathcal{T}_U needs to be done by an administrator. Concretely the mapping is to assign a symbolic region to each node or group of nodes.
- The time should also be mapped to a WSN friendly value. Human readable expressions like days or hours have to be expressed in number of epochs for the WSN query. The sink knows the duration of an epoch and when t_0 occurred, hence the translation for time is a trivial task.

4 Query Translation to TinyPEDS

For a better understanding of the introduced query translation model, we exemplarily illustrate the translation from (q, t_U) to $\varpi \in \mathcal{M}$ and \mathcal{M} representing the *tinyPEDS* semantic. We elaborated a user-friendly SQL-like language very similar to *tinyDB* [18]. Our *tinyPEDS* query syntax is as follows:

```
SELECT agg<expr> | <expr>, ...
[FROM sensors | <expr>]
[WHERE <pred>]
[TIME [BETWEEN <expr> AND <expr> | LAST <expr>]]
[GROUP BY <expr>]
[HAVING <pred>]
```

One can notice that the FROM term is optional. When not set, the default value is 'FROM sensors'. This language aims to provide the most functionality to the user, though it is limited compared to the full SQL language set. There are for example no nested queries. Furthermore, our language should be easily extendable to synchronous WSNs queries. By having a similar syntax to the most used synchronous WSN database, which is *tinyDB*, we ensure larger acceptance among end-users. The essential differences between *tinyDB* and *tinyPEDS* are discussed in the Annex A.

As an example of user-friendly (SQL-like) query $(q, t_u) \in \mathcal{Q} \times \mathcal{T}_U$:

```
SELECT AVG(light), room FROM sensors
WHERE room = 'kitchen' OR room = 'library'
TIME BETWEEN '12:00' AND '13:00'
GROUP BY room
```

The format of a “WSN friendly” *tinyPEDS* flooding message is $\langle region, duration, aggregation, TTL, QT \rangle$. Our model supports hierarchical topologies and here *region* is any subregion of the WSN. In *tinyPEDS*, typically *region* can be noted as $region = \langle level_1 \triangleright level_2 \dots \triangleright level_l \rangle$ with $level_i \in \mathcal{P}(ALL)$ and $ALL := \{\mathcal{Q}_{(i,1)}, \dots, \mathcal{Q}_{(i,4)}\}$ for $1 \leq i \leq l$. As a consequence, the smallest unit of an area to which a symbolical location can be mapped is one of the lowest hierarchical subregion of the WSN. Duration is any subinterval over the

WSN’s current lifetime, aggregation describes the mode of in-network processing, which can be the addition or comparisons operations. The parameter TTL is the time-to-live and QT describes the query type which can be either “continuous (C)” or “disaster (D)”. The problem of mapping the SQL-like query to the WSN query is further discussed in Section 5. A controlled flooding message $\varpi = \langle region, [t_x, t_y], aggregation, ttl_{max}, C \rangle$ is handled by receiving sensors $s \in \mathcal{N}$ as denoted in Algorithm 1. In *tinyPEDS*, data monitored is stored at the aggregator node itself, and at one of its neighbouring aggregator. Notice that the query model for distributed database entries of the WSN is different prior and after a disaster strike. After a disaster, where a large number of nodes have died in a limited area, the query is flooded to the complementary region of where the data was originally monitored, as described in Algorithm 2. Thus a disaster query would hopefully harvest the monitored data of the dead nodes in their backup nodes, it is only issued when a continuous query has failed. With the data harvested from the continuous and disaster query responses, one can reconstruct the data completely.

For the concrete setting of $\varpi = \langle \mathcal{Q}_{(1,1)} \triangleright \mathcal{Q}_{(2,2)} \triangleright \mathcal{Q}_{(3,1)} \cup \mathcal{Q}_{(2,4)} \triangleright \mathcal{Q}_{(3,1)}, [t_x, t_y], +, 20, C \rangle$ and the WSN characteristics listed in table 1, ϖ translates into the controlled flooding pattern depicted in the figure 1 of our the GloMoSim simulation. $\mathcal{Q}_{(x,y)}$ denotes a subregion (quarter) of \mathcal{N} , whereas x ($1 \leq x \leq l$) stands for the quarter’s hierarchy level and y ($1 \leq y \leq \omega$) identifies the ω quarters in a cycled order of the corresponding hierarchy level l .

Algorithm 1 *Continuous Database Query* /* for any receiving sensor node $s \in \mathcal{N}$ */

```

if  $s \in \mathcal{Q}_{(x,y)}$  AND  $\mathcal{Q}_{(x,y)} \subseteq pathTo(region)$  then
  if  $ttl_{current} > 1$  then
     $ttl_{current} = ttl_{current} - 1$ 
     $s \rightarrow * : \langle region, [t_x, t_y], aggregation, ttl_{current}, C \rangle$ 
    if  $aggregation = true$  AND  $storage_{[t,t+1]} \cap region \neq \emptyset$  AND  $t_x \leq t \leq t_y$  then
       $s \rightarrow R : \langle storage_{[t,t+1]} \rangle$ 
    end if
  end if
else
   $ttl_{current} = 0$ 
end if

```

5 Problem Statement

We are now in the position to formulate the problem statement of this work: Given a $\mathcal{T} = (\mathcal{T}_U, \mathcal{T}_N)$ for a \mathcal{M} that ensures that an adequate $f^* : \mathcal{Q} \times (\mathcal{T}_U, \mathcal{T}_N) \rightarrow \mathcal{M}$ exists, how to make the originator of any $q \in \mathcal{Q}$ verifiable for the receiver of $\varpi \in \mathcal{M}$ over the detour of f^* (and not f)?

Algorithm 2 *Disaster query*

```

if  $s \in \mathcal{N} \setminus \mathcal{Q}_z$  then
  if  $tll_{current} > 1$  then
     $tll_{current} = tll_{current} - 1$ 
     $s \rightarrow * : \langle region, [t_x, t_y], aggregation, tll_{current}, D \rangle$ 
    if  $aggregation = true$  AND  $storage_{[t, t+1]} \cap region \neq \emptyset$  AND  $t_x \leq t \leq t_y$  then
       $s \rightarrow R : \langle storage_{[t, t+1]} \rangle$ 
    end if
  end if
else
   $tll_{current} = 0$ 
end if

```

WSN size, quadrant size	400x400, 50
num. nodes	240-407
node's transmission range	50
hierarchy levels (l)	3
num. quarters per level (ω)	4
radio layer	CSMA
propagation pathloss	two-way

Table 1. GloMoSim simulation parameters.

\mathcal{T}_U	...	library	kitchen	...
\mathcal{T}_N	...	$\mathcal{Q}_{(2,2)} \triangleright \mathcal{Q}_{(3,1)}$	$\mathcal{Q}_{(2,4)} \triangleright \mathcal{Q}_{(3,1)}$...

Table 2. Concrete topology setting for $\mathcal{T} = (\mathcal{T}_U, \mathcal{T}_N)$.

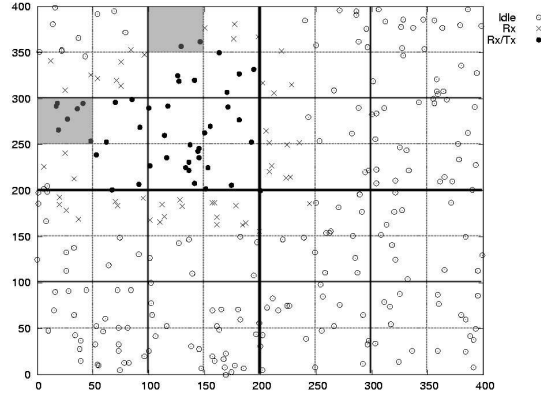


Fig. 1. Controlled flooding of the continuous database query $\langle \mathcal{Q}_{(1,1)} \triangleright \mathcal{Q}_{(2,2)} \triangleright \mathcal{Q}_{(3,1)} \cup \mathcal{Q}_{(2,4)} \triangleright \mathcal{Q}_{(3,1)}, [t_x, t_y], +, 20, C \rangle$ in a WSN with $l = 3$ and $\omega = 4$.

The precondition of our problem statement addresses the fact that for a semantical appropriate interpretation of q one has to provide additional information t_N to translate q in a corresponding ϖ . However, to still ensure a user-friendly query process we must hide t_N from the end-user, namely t_N must be incorporated into the query process *after* the user has formulated his query. This implies that f^* can only be applied after q has been formulated by the user. This observation has different impact for the previously introduced different WSN types:

a) *asynchronous WSN*: In this setting the functionalities of the reader and the sink node are co-located. The end-user formulates a query (q, t_U) . However, t_N is available at the sink node respectively the reader and it is due to the query translation framework to translate (q, t_U) to ϖ :

$$\begin{array}{ccc} \text{reader/sink} & & \text{sensor} \\ (q, t_U) & & \\ \downarrow & & \\ f^*(q, t_U, t_N) = \varpi & \rightarrow & \varpi \end{array}$$

b) *synchronous WSN*: In this setting the reader device and the sink node are two separate units. A multitude of reader devices can access the WSN from anywhere in the core network via the sink node. In such a setting it is unrealistic to assume that \mathcal{T}_N is known to each reader device. Instead \mathcal{T}_N should solely be located at the sink node. Therefore, we assume the following setting:

$$\begin{array}{ccc} \text{reader} & \text{sink} & \text{sensor} \\ (q, t_U) \rightarrow f^*(q, t_U, t_N) = \varpi & \rightarrow & \varpi \end{array}$$

Our problem statement addresses the security problem that arises due to the fact that t_N is subsequently incorporated into a query (q, t_U) *after* the end-user entry. How to ensure query message originator verification between the end-user and the verifying sensor nodes on (q, t_U) although sensor nodes finally receive ϖ ? Note that this is of highest relevance for *synchronous WSNs* in which the reader device and the sink node are separate units.

6 Truncated Hash Chains

We introduce the notion of dedicated security primitives and define a hash-chain, which is also known as Lamport's hash-chain [9], as $x_{i+1} = h(x_i)$ with x_0 being the anchor and h being an un-keyed one-way hash function which maps bit-strings to t bits. One-wayness means that given x , the image $y = h(x)$ can be easily computed, but the other way around, that is given y finding a pre-image, is difficult. We denote $x_i = h^i(x_0) = h(\dots(h(x_0))\dots)$. We define $x_{i+1}^{[1, t-k]}$ to be a $(t-k)$ -truncated hash value from a t -bit secure hash function: $x_{i+1}^{[1, t-k]} \leftarrow h(x_i)$. W.l.o.g. $[1, t-k]$ indicates that $x_{i+1}^{[1, t-k]}$ consists out of the first $t-k$ bits of $h(x_i)$.

7 Access Control for synchronous WSNs

We propose an access control protocol for (synchronous) WSNs which

1. supports the mapping from user-friendly queries to a WSN architecture appropriate flooding message,
2. hides the WSN's master secret from the set of reader devices,
3. supports newcomers to the set of reader devices, and
4. securely links the access protocol part from the WSN with the protocol part from the infrastructure.

Topic 1) ensures that in particular a message flow as described in Section 5b) is supported. Topic 2) ensures that readers cannot directly send queries to the WSN, namely skip the sink node. In addition, the set of the reader devices may be huge and the knowledge of each client about the master secret would be unacceptable.

Transmitting:	Processing:
1. $R \rightarrow S : (q, t_U) H_{x_{n-i}}(q, t_U)$	$S : x_{n-i} = h^{n-i}(x_0)$ $H_{x_{n-i}}(q, t_U) \stackrel{?}{=} H_{x_{n-i}}(q, t_U)$ $x_{n-i-1} = h^{n-i-1}(x_0)$ $f^*(q, t_U, t_N) = \varpi$
2. $S \rightarrow R : \varpi, z_0, H_{x_{n-i-1}}(x_{n-i}, \varpi)$	$R : x_{n-i-1} = h^{n-i-1}(x_0)$ $H_{x_{n-i-1}}(x_{n-i}, \varpi) \stackrel{?}{=} H_{x_{n-i-1}}(x_{n-i}, \varpi)$ $y_{m-j} = z_0 \oplus x_0$
3. $R \rightarrow S : \varpi, H_{x_{n-i-1}}(x_{n-i}, y_{m-j}, \varpi)$ $H_{y_{m-j}}(\varpi)$	
4. $S \rightarrow s : \varpi, H_{y_{m-j}}^{[1, t-k]}(\varpi)$	$s : y_{m-j} = h^{m-j}(y_0)$ $H_{y_{m-j}}^{[1, t-k]}(\varpi) \stackrel{?}{=} H_{y_{m-j}}^{[1, t-k]}(\varpi)$

Table 3. WSN access control protocol for the i -th query of the reader device.

Compared to the scarce radio link connecting the sink node and the sensor nodes we assume the bandwidth requirements between the reader device and the sink node to be rather relaxed. We further assume the sink node to be tamper-resistant and to provide enough storage space to administrate a list of all end-users (readers) who are allowed to access the WSN. For such a setting one solution for a user-friendly and DoS resilient access control protocol to the WSN may be based on

- a challenge-response based on a keyed one-way function,
- two hash chains, and
- a truncated (keyed) hash function.

The challenge response protocol runs between the reader device (R) and the sink node (S) whereas we solely apply a truncated hash function on the scarce radio link between S and a sensor (s). This ensures a weak but still reasonable secure query message authentication at the sensor nodes to prevent DoS attacks from “any” unauthorised reader which simply circumvents the sink node S . Without such a mechanism it would be possible to cheat and to directly address a query request to the WSN. We apply two hash chains. Hash chain number one which is generated based on a secret anchor x_0 is known by the sink node and one particular reader device. The sink node stores for each particular reader a different pair (x_0, n) whereas each reader stores the secret (x_0, n) . The value n represents the number of iterations of the applied hash function, namely x_n is the hash value $x_n = h^n(x_0)$. The hash value x_n is public and reveals no secret. Hash chain number two is generated based on the secret anchor y_0 . The tuple (y_0, m) is known by the sink node and each sensor node in the WSN. This tuple is stored by the manufacturer within the sensor nodes preliminary to the roll-out of the WSN. Again $y_m = h^m(y_0)$ is public not revealing any secret. Both hash chains are applied in a simple yet efficient way to control access to an asynchronous WSN. The basic idea is to apply intermediate hash values $x_i = h^i(x_0)$ with $1 \leq i \leq n$ and $y_j = h^j(y_0)$ with $1 \leq j \leq m$ to keyed hash functions $H_{x_i}(\varpi)$ respectively $H_{y_j}(\varpi)$, namely message authentication codes (MAC). The $(n - i)$ -th hash value which can be derived from x_0 is used by the particular reader for its i -th query addressing the WSN. The $(m - j)$ -th value which can be derived from y_0 represents the j -th request of *any* reader device. However, due to the extreme bandwidth limitation on the radio (IEEE 802.15.4 WPAN) and due to the fact that the number of transmitted bits directly translates into a reduced lifetime of the WSN we propose to use $(t - k)$ -truncated keyed hash values $H_{y_{m-j}}^{[1, t-k]}(\varpi) \leftarrow H_{y_{m-j}}(\varpi)$. The full protocol is described in table 3.

Remarks:

- the secret (x_0, n) is known by the sink node and a specific reader device. The secret (y_0, m) in particular is not known by the reader devices. It is exclusively stored at the sensor nodes and at the sink node. To still enable a reader device to compute $H_{y_{m-j}}(\varpi)$ and to know y_{m-j} , in Step 1 the sink node performs the \oplus function (bitwise addition modulo 2) on x_0 and the actual WSN secret y_{m-j} . In Step 2) the reader device bitwise adds x_0 to z_0 to compute the actual WSN key which is valid for the actual flooding message ϖ .
- if everything works well, it is not really necessary to provide R the knowledge of y_{m-j} . But as the experience shows, it is rather unrealistic to assume that the sink S can always communicate with each sensor s . For example, in huge WSN, intermediate nodes are necessary to transport a message from S to s . If this connection is broken, R can use the knowledge of y_{m-j} to get directly in contact with s . As s updates the value of y_{m-j} to y_{m-j-1} for the next flooding message, y_{m-j} provides only a kind of temporary permission to get information from s , what limits the damage R could cause.

- any $s \in \mathcal{N}$ where $H_{y_{m-j}}^{[1,t-k]}(\varpi) \stackrel{?}{=} H_{y_{m-j}}^{[1,t-k]}(\varpi)$ is true, locally broadcasts the message ϖ , $H_{y_{m-j}}^{[1,t-k]}(\varpi)$, otherwise it drops the query.
- the problem of message loss over the wireless can be handled at any $s \in \mathcal{N}$ by also validating if $H_{y_{m-j}}^{[1,t-k]}(\varpi) \stackrel{?}{=} H_{y_{m-j-2}}^{[1,t-k]}(\varpi)$ in case the check $H_{y_{m-j}}^{[1,t-k]}(\varpi) \stackrel{?}{=} H_{y_{m-j-1}}^{[1,t-k]}(\varpi)$ failed. If the second check succeeds s forwards the query and sets $j = j - 1$.
- only $t - k$ additional bits over the scarce wireless are needed. An adversary who eavesdrop the value $H_{y_{m-j}}^{[1,t-k]}(\varpi)$ might misuse this knowledge to replace ϖ by his own flooding message. However, as the value y_{m-j} is secret, the problem is to find a value $\varpi' \neq \varpi$ such that $H_{y_{m-j}}^{[1,t-k]}(\varpi) = H_{y_{m-j}}^{[1,t-k]}(\varpi')$ for an unknown value y_{m-j} . This is also known as a *target collision*. We expect a hash function suitable for a hash chain to be *target collision resistance*. Note that it is widely believed that for $t = 80$ finding a target collision is as hard as factoring an RSA modulus of 1024-bits. On the Berkeley notes e.g. RC5 based hash chains and MACs are quite competitive in terms of speed (2.22-4.18ms), code size (1738 Bytes) and data size (136 Bytes).

8 Proposed Approach for Data Concealment in WSNs

Access control for WSNs is only valuable if data concealment over the wireless is provided. Approaches like TinySec with RC5 respectively Skipjack or IEEE 802.15.4 WPAN's AES-CTR provide link layer security in a hop-by-hop manner. However, since query response traffic is characterised by in-network processing and data aggregation of reverse multicast traffic each aggregating node must decrypt all receiving data, aggregate the data and subsequently encrypt the aggregated value again. This is suboptimal due to i) a lack of security at the aggregating nodes and ii) due to the energy waste for multiple decryption and encryption operations. Therefore in [15] we propose for the aggregation functions *sum*, *average*, *movement detection* and *variance* to apply a *symmetric/asymmetric additively privacy homomorphism (PH_S)* to conceal reverse multicast traffic end-to-end. A *PH_S* is an encryption transformation which has the property

$$a + b = D_k(E_k(a) + E_k(b)) \quad (1)$$

where a and b belong to the plaintext domain and E and D are encryption respectively decryption transformations on the key k . Using such an *PH_S* in WSNs means that an aggregator node can perform above aggregation functions on incoming ciphertexts. With conventional encryption schemes it would need to decrypt incoming ciphers before performing the aggregation function.

9 Related Work

The first work in progress report investigating the problem of authenticated querying in sensor networks appeared in 2005 from Benenson [2]. Benenson intro-

duced the problem of node querying and analysed the design space for authenticated queries in WSNs. Although Benenson describes techniques for authenticated querying she does not provide a concrete solution to the problem of authenticated queries. Recently a concrete solution to this problem followed in [3]. It is based on the idea of using 1-bit MACs per sensor node. A sensor node receiving a query can infer with probability $1/2$ if the query stems from an authenticated reader device.

Zhou and Ravishankar [19] have proposed the use of dynamical Merkle trees and one-way hash chains in order that the sensors are able to authenticate mobile sinks. The mobile sinks must get for each activity the necessary credential from the base station, so that they can then locally query the sensors. The sensors can then verify the authenticity of mobile sinks by just storing the prior knowledge of the root of the Merkle tree. Contrary to the work at hand, these approaches are focused on the WSN and not on the fixed network.

More generally, a set of authentication approaches for restricted devices have been proposed. We restrict ourselves by referring to the resurrecting duckling approach from Stajano and Anderson [12], the Guy Fawkes protocol [1], the TESLA approach from Perrig et al. [11], and the Zero Common-Knowledge (ZCK) protocol [14].

10 Conclusion

We introduced the problem of end-user friendly WSN queries and DoS resilient access control to WSNs. We propose to use an access control protocol which is based on a challenge-response protocol and truncated hash values over the scarce wireless to overcome the introduced problems. The access control mechanism provides two way mutual authentication between the reader device and the sink node as well as a lightweight query authentication at the sensor nodes. The latter is mandatory to prevent un-authorized users from flooding query messages to the WSN by circumventing the sink node. The proposed data storage architecture on the WSN side is tinyPEDS which ensures an encrypted distributed data storage.

Acknowledgment

We would like to thank Axel Poschmann and Andre Weimerskirch for their input to Section 2. We would also like to thank Alban Hessler who gave input to the camera ready version of this paper.

A Comparing tinyPEDS with tinyDB

We argued in Section 4 that the *tinyPEDS* front-end language is very similar to the one of *tinyDB*, because we do not want the user to learn a new query syntax for each WSN database. However, one has to keep in mind that the underlying databases and mechanisms are fundamentally different. Due to space restrictions

	<i>tinyPEDS</i>	<i>tinyDB</i>
WSN type	Asynchronous	Synchronous
Storage policy	In-network	At sink Some storage points possible
Dissemination model	Controlled flooding	Multicast (tree routing)
System security	Encrypted storage End-to-end confidentiality	None

Table 4. Databases features.

we only sum up the main differences. They are listed in table 4:

The data model has the same shape for both WSN databases. We can see the WSN as one table, i.e. *sensors*, which has a column for each type of data, e.g. temperature; and a row for:

- each node and interval of time in *tinyDB*
- each lowest subregion and epoch in *tinyPEDS*

One major difference is that *tinyDB* uses acquisitional queries, meaning that records of the table are only materialized as needed to satisfy a query and subsequently stored for a short period of time or delivered out of the network. Consequently queries in *tinyDB* only concern present or future values. In opposite to *tinyPEDS*, where queries harvest data of the past thanks to the in-network persistent storage.

References

1. R. Anderson, F. Bergadano, B. Crispo, J.-H. Lee, C. Maniavas, R. Needham, “A New Family of Authentication Protocols”, *ACM Operating Systems Review*, 1998.
2. Z. Benenson “Authenticated Queries in Sensor Networks”, *Second European Workshop on Security and Privacy in Ad Hoc and Sensor Networks, ESAS’05, LNCS 3813*, pp. 54-67, Visegrad, Hungary, 2005.
3. Z. Benenson, L. Pimenidis, F.C. Freiling, S. Lucks “Authenticated Query Flooding in Sensor Networks”, *4th IEEE Conference on Pervasive Computing and Communications Workshops*, pp. 644-647, Pisa, Italy, 2006.
4. FIPS PUB 140-2, “Security Requirements for Cryptographic Modules”, *Federal Information Processing Standards Publication*, National Institute of Standards and Technology.
5. W. Heinzelmann “Application-Specific Protocol Architectures for Wireless Networks”, *PhD thesis*, MIT, 2000.
6. J. Girao, D. Westhoff, E. Mykletun, T. Araki, “TinyPEDS: Persistent Encrypted Data Storage in Asynchronous Wireless Sensor Networks” to appear as regular paper in *Elsevier Ad Hoc Journal*.
7. J. M. Hellerstein, W. Hong, S.Madden, K. Stanek, “Beyond Average: Towards Sophisticated Sensing with Queries”, In *Workshop on IPSN’03*, Palo Alto, CA, USA, April 2003.
8. H. Karl, A. Willig, “Protocols and Architectures for Wireless Sensor Networks”, *Wiley*, 2005.

9. L. Lamport, "Password authentication with insecure communication", *Commun. ACM* 24(11), pp. 770–772, ACM Press, New York, NY, USA, 1981.
10. S. Madden, M. J. Franklin, J. Hellerstein, W. Hong, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks" In 5th Symposium on OSDI, 2002.
11. A. Perrig, R. Canetti, J.D. Tygar, D. Song, "The TESLA broadcast authentication protocol", *RSA CryptoBytes*, 5(Summer), 2002.
12. F. Stajano, R. Anderson, "The resurrecting duckling: Security issues for ad hoc wireless networks", 7th International Workshop, volume 1796 of *Lecture Notes in Computer Science*, pp. 172-194. Springer-Verlag, Berlin Germany, 2000.
13. S. Tilak, N. B. Abu-Ghazaleh, W. Heinzelmann, "Collaborative Storage Management in Sensor Networks", *Journal of Ad Hoc & Ubi. Comp.*
14. A. Weimerskirch, D. Westhoff, "Zero-Common Knowledge Authentication for Pervasive Networks", 10th Selected Areas in Cryptography, SAC'03, Springer-Verlag LNCS 3006, pp. 73-87, Ottawa, Ontario, CA, August 2003.
15. D. Westhoff, J. Girao, M. Acharya, "Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks: Encryption, Key Pre-distribution and Routing", in *IEEE Transactions on Mobile Computing*, Vol. 10, October 2006.
16. D. Westhoff, "End-user friendly and DoS Resilient Access Control for WSNs", 13th International Conference on Telecommunication, ICT, Portugal, May 2006.
17. J. Girao, D. Westhoff, M. Schneider, "CDA: Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks", In *IEEE International Conference on Communications (ICC'05)*, Seoul, Korea, May 2005.
18. Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong, "TinyDB: an acquisitional query processing system for sensor networks", *ACM Trans. Database Syst.*, Vol. 30 No 1, 2005
19. Li Zhou, Chinya V. Ravishankar, "Dynamic Merkle Trees for Verifying Privileges in Sensor Networks", In *IEEE International Conference on Communications (ICC'06)*, Istanbul, Turkey, June 2006