A Secure and Resilient WSN Roadside Architecture for Intelligent Transport Systems

Jens-Matthias Bohli, Alban Hessler, Osman Ugus, and Dirk Westhoff
NEC Laboratories Europe, Network Research Division
NEC Europe Ltd.
Kurfürsten Anlage 36
69115 Heidelberg, Germany
{bohli,alban.hessler,osman.ugus,dirk.westhoff}@nw.neclab.eu

ABSTRACT

We propose a secure and resilient WSN roadside architecture for intelligent transport systems which supports the two complementary services accident prevention and post-accident investigation. Our WSN security architecture is stimulated by the understanding that WSN roadside islands will only be rolled-out and used when hardware costs are close to the minimum. We provide a purely software based security solution which does not rely on costly HW components like road side units (RSU) or tamper resistant modules on sensor nodes. We use existing components, but also describe protocols that may be of independent interest.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Applications*

General Terms

Algorithms, Design, Experimentation, Security

Keywords

wireless sensor networks, vehicular ad-hoc networks, security

1. INTRODUCTION

We foresee that in the near future, two types of wireless networks will operate in an integrated manner aiming at an increased level of public safety and liability; *Vehicular Ad Hoc Networks* (VANET)s and *Wireless Sensor Networks* (WSN)s.

On the one hand vehicular-to-vehicular communication within a VANET is close to reality: The upcoming RF standard IEEE 802.11p as well as the actual DSRC channel allocations in the higher 5.8 GHz band for various public safety services clearly indicate the next step towards a real civilian usage of VANETs. Hereby, we point out that for the VANET adapted secure WSN middleware architecture proposed in this work it is insignificant whether car-to-car communication is indeed multi-hop, e.g. based on position-based routing like Greedy Perimeter Stateless Routing (GPSR) with a position service like GPS/Galileo, or whether the communication is envisioned to be in a single-hop manner and basically purely relying on local broadcast.

Complementary to the pure C2C communication, roadside to car (R2C) services are also currently discussed within the relevant IVC consortia *Car to Car Communication Consortium* (C2CCC), *Vehicular Safety Communication Consortium* (VSCC) and the *Internet ITS Consortium*. In this work, we propose and analyse a cost-efficient and practicable R2C approach based on Wireless Sensor Networks (WSN)s. We foresee that soon integrated road sensors will be used. Many WSN islands could be rolled-out on the road surface or at the road boundary typically at curves, tunnels and bridges, and even on a much wider scale. They can be used to measure data like humidity, temperature, light or detect movement to compose higher safety and liability services.

Such an integrated vehicular and WSN roadside architecture could be used for the provision of the two complementary services

- accident prevention, and
- post-accident investigation.

To support *accident prevention*, roadside sensor nodes measure the road condition at several positions on the surface, aggregate the measured values and communicate their aggregated value to a passing vehicle. The vehicle generates a warning message and distributes it to all vehicles in a certain area, e.g. by using a specific form of georouting, namely geocast. One can even imagine chargeable *premium services*. Services like velocity assistance and recommendations or more generally infrastructure-based traffic control may be attractive for the driver. They increase the driver's comfort without causing emergency situations in the absence of such services. Technically, one could provide such services by piggybacking WSN roadside information of a WSN far ahead via a vehicle driving in the opposite direction and downloading it to a near-by WSN for the oncoming traffic.

Note that we recommend a setting where each vehicle is equipped with an *on board-unit* (OBU) containing two RFs; namely IEEE 802.11p and IEEE 802.15.4. We propose such an architecture for the striking reason that it dramatically

WiSec'08, March 31–April 2, 2008, Alexandria, Virginia, USA.

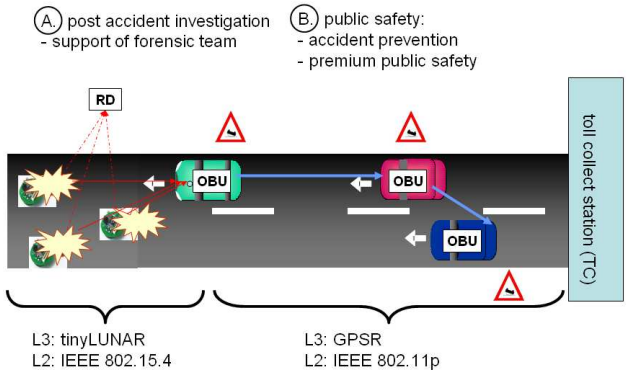

Figure 1: Overall WSN roadside architecture for Intelligent Transport Systems.

reduces the costs for the WSN islands. Under such a setting, no *roadside units* (RUs) other than the sensor nodes themselves are required. Consequently, such an architecture helps ensuring the minimum possible costs for the WSN islands. We foresee that the WSN hardware, roll-out and maintenance costs are a crucial success criterion for a real WSN island penetration in the context of vehicular communication since the roll-out of WSN islands with high probability is an investment of a single provider or only a few providers. Reducing the costs for a WSN roadside island also motivates our decision not to use tamper resistant modules on the sensor nodes with the consequence to be forced providing the best achievable security by purely applying software solutions and a proper architectural design.

We argue that having a single additional antenna in each vehicle will only negligibly effect the vehicle's end-price. Recent measurements [23] with an omnidirectional RF IEEE 802.15.4 antenna show that up to 50 packets can be received at the vehicle, assuming a velocity in the range of 70km/h[1]. Note, that although both RF technologies are (currently) operating in the 2.4GHz band, in principle a simultaneous usage of IEEE 802.15.4 and IEEE 802.11x is possible without causing mutual interferences [20].

To support *post-accident investigation*, sensor nodes continuously measure the road condition and store this information within the WSN itself. Storing the road condition over the long run may be of interest for a forensic team. In contrast to the accident prevention service, such a liability service will be limited to a well specified group of end-users, e.g. insurance companies or the road patrol. Information stored within the WSN will be helpful to judge a driver's driving style according to the road condition at the moment of an accident. We point out that the post accident investigation service does not require any VANET communication. An IEEE 802.15.4 enabled reader device (RD) allowing authorized queries to the WSN roadside island is adequate. Therefore, this service can be a stimulating activity ensuring the critical mass during the penetration phase of a full fledged ITS architecture for vehicles.

[1]With the usage of *directed* IEEE 802.15.4 antenna in the vehicle higher velocities will be supported.

The overall WSN roadside architecture for the two complementary services accident prevention and post accident investigation is illustrated in Figure 1.

Our Contribution: It is the contribution of this work to provide a middleware architecture for securing WSN roadside islands. Security solutions are specifically adapted to the requirements for the complementary services accident prevention and post-accident investigation. We think that the protocols for access control and resilient data aggregation are innovative enough to be of independend interest. We demonstrate that the architecture is secure, robust and offers the service for a considerable lifetime. The latter we investigate by applying our code to the AVRORA simulation and analysis framework [21].

2. ADVERSARIAL MODEL AND SECURITY REQUIREMENTS

We assume the adversary is in complete control of the wireless channel and an arbitrary number of sensor nodes. The attacker can eavesdrop data over the wireless broadcast medium (passive attack), control the communication channel to catch, destroy, modify and send data (active attack), or corrupt a sensor node. If the adversary controls a sensor node, she gains knowledge of all the sensitive information stored at this node (physical capture). This is what we call *WSN adapted* Dolev-Yao style adversary, who focuses more on malicious protocol participants than the standard Dolev-Yao adversary [9].

We observe that the security requirements for *accident prevention* and *post accident investigation* at the WSN roadside of an ITS system are fundamentally different:

- **Accident prevention:** the WSN needs to send the actual monitored road condition to a vehicle whenever one passes. The basic security requirements are i) a plausibility check for the aggregated value to mitigate the effect of bogus sensor readings (*stealthy attack*), and ii) real-time access control in a mobility scenario in case of the provisioning of premium services. In case of supporting premium services one can even imagine applying end-to-end confidentiality within the WSN.

- **Post-accident investigation:** only when the WSN receives a query from an authorized party, e.g. a member from a forensic team, it provides measured and aggregated data from the past. The basic security requirements besides a plausibility check are i) a time-uncritical access control for a very restricted set of entities in a static setting, and ii) data confidentiality as well as storage of data replica for the sensed and stored data.

A deep discussion on security requirements on the VANET side is not considered in this work. However, one could combine the architecture proposed in this work with the work on secure incentives in VANETS [14] to stimulate cooperative forwarding behavior within the VANET. One approach to combine the contradicting security requirements privacy and non-repudiation in the context of IVC may be based on the usage of group signatures as proposed in [1]. A good overview on how to secure vehicular and ad hoc networks is given in [18].

3. A BIRD'S EYE VIEW: SECURE AND RELIABLE WSN ROADSIDE ARCHITECTURE

At first, we introduce the available components for ensuring WSN roadside security by name and functionality before they are described in more detail in the following Sections. We show how these and new components can be adapted to the introduced ITS scenario.

3.1 Networking Components

We apply aggregated data transport in the WSN. The roadside WSN is structured in clusters consisting of several collocated nodes that measure a correlated quantity. Each cluster contains an aggregator node that aggregates the values of its cluster. Depending on the number of clusters, the cluster values get aggregated on several levels to build up the common value for the WSN. The basic networking components for routing and aggregator node election within the WSN island that we consider are *tiny Lightweight UNderlay AdHoc Routing* (tinyLUNAR) [17] and *Secure Aggregator Node Election* (SANE) [19]. TinyLUNAR is a layer 2.5 protocol that takes benefits from an extended label-switching forwarding technology. SANE is a protocol that flatly balance the energy consumption within the WSN by electing aggregator nodes per epoch anew. A focus of SANE is also on insider attacks during the election process. We will also apply SANE for the temporary and random election of a witness node as we explain later.

Please note that, besides the motivation to flatly balance the energy consumption within the WSN, also in relatively small WSNs like the considered roadside WSN islands it makes sense to elect aggregator nodes for the following reasons: i) aggregator nodes are privileged to store data replica, e.g. from neighboring clusters, ii) it is envisioned that only the 'master' aggregator node per epoch communicates with an automotive.

3.2 Securing the Accident Prevention and Premium Services

3.2.1 Accident Prevention Service

For securing the *accident prevention* service, we propose to use an adaptation on *RANSAC-Based Resilient Aggregation in Sensor Networks* (RANBAR) [5]. It provides a outlier elimination technique to resiliently aggregate data values. The algorithm is based on the random sample consensus. However, RANBAR assumes that the aggregator nodes themselves are trustworthy. In our ITS adapted setting RANBAR either runs on a sensor node or on the vehicle's OBU. Note that, since RANBAR is operating on plaintext sensed values it is not suited to be combined with an end-to-end encryption scheme.

To also deal with cheating aggregator nodes, one could apply a provably secure framework for information aggregation in WSNs [15], [8]. Unfortunately, the higher system security comes at the costs of (unacceptable high) transmission costs. To reduce the required transmission costs, we are voting for a witness based approach [10] with two modifications: firstly the witness node for an aggregator node is *randomly* elected. Secondly, in similarity to the hierarchical operation of the aggregator nodes itself also the witness nodes are structured hierarchical.

3.2.2 Premium Services

In case the operator is aiming at a differentiation of basic automotive safety services and premium automotive services, one could offer the latter protected against unauthorized use. This includes an *access control* that is based on multiple 1-bit truncated keyed hash values proposed by Canetti [6] and adapted to WSNs by Benenson et al. [4]. Each node randomly stores a subset of keys being able to verify a subset of truncated keyed hash values generated and transmitted from a passing vehicle. Another component is encrypted data aggregation based on a symmetric homomorphic encryption transformation named *Concealed Data Aggregation* (CDA) [24]. CDA provides end-to-end encryption by at the same time ensuring in-network processing. A suitable CDA derivate for the requirements of a WSN to VANET scenario is presented in Section 4.1.6.

3.3 Securing the Post-Accident Investigation Service

To ensure a secure and reliable post-accident investigation, we apply the *tiny Persistent Encrypted Data Storage* (tinyPEDS) [11] with access control and query mapping similar as it is proposed in [2]. The benefits of tinyPEDS are manifold: 1) stored data are encrypted and even the storing node cannot decrypt the ciphered values, 2) transmission costs for collaborative and distributed data storage are minimised, 3) persistent storage space is balanced over multiple sensor nodes, and, 4) nodes know from which region and for which epoch they store data. However, they do not know the values they are storing. Since nodes may disappear over time, a replicated and read-protected, but yet space- and energy-efficient, data storage is mandatory. TinyPEDS ensures the encrypted storage of the environmental fingerprint for asynchronous wireless sensor networks over the time and region. Even if parts of the network are exhausted, restoring rules ensure that, with a high probability, environmental information from past is still available to the forensic team. Furthermore, since tinyPEDS needs to perform data aggregation on encrypted data we apply an additively homomorphic encryption transformation. For the WSN operating in asynchronous mode our choice was an asymmetric homomorphic encryption transformation, namely EC-ElGamal with a suitable mapping function. A space (and execution time) optimized implementation for an 8-bit processor is available [22]. Note that for the encryption purely the public key is stored within the WSN. One can also consider a nested arrangement of symmetric and asymmetric homomorphic encryption transformation. In that case we propose to use the streamcipher based encryption scheme from Castelluccia et al. [7] for encrypted transmission within a WSN cluster plus the EC-ElGamal encryption for long term storage within the roadside WSN.

The security and reliability components listed above are implemented in nesC and are available as tinyOS modules. Except the EC-ElGamal implementation which only runs on an 8-bit processor, all modules run on the reference platforms Mica2, MicaZ and Telos Sky. The memory footprint for selected software pieces is documented in Table 1.

3.4 Key Setting

The security modules we propose for a secure and resilient WSN roadside architecture assume the storage of various keys respectively seed values at different locations. We dif-

Software item	RAM [kB]	ROM [kB]
tinyLUNAR	1.7	3.4
tinyPEDS	0.5	1.0
EC-ElGamal	0.7	4.4

Table 1: Memory footprint for selected software pieces, compiled for the Atmega128 CPU.

module	TC	RD	OBU	N_i
use case 1: public accident prevention				
authenticity (Sec. 4.1.2) (Sec. 4.1.5) RANBAR (Sec. 4.1.1) SANE (Sec. 4.1.4)	$\mathcal{K}$		$\mathcal{K}^{ID}$	$\mathcal{K}^i$, $\mathcal{X}_i$ $G, p_i^{(0)}$
use case 2: chargeable premium service				
access control (Sec. 4.1.3)	$\mathcal{K}$		$\mathcal{K}^{ID}$	$\mathcal{X}_i$
CDA (Sec. 4.1.6)			$\{\hat{K}_C\}_C$	$\hat{K}_{C(i)}$
use case 3: post-accident investigation				
tinyPEDS (Sec. 4.1.7)		K_p		K_q
access control		K_p		K_q

Table 2: Required key and seed setting for the provided services.

ferentiate between a storage at a *sensor node* N_i, a vehicle *on board-unit* (OBU) with identity ID, a *reader device* (RD) of the forensic team, and finally the operator, represented by the *toll collect station* (TC).

For a security architecture purely focusing on the use case *public accident prevention*, the proposed security modules require keys for the authentication inside the WSN and the convergecast authentication. The WSN operator stores a set $\mathcal{K}$ of master keys $k_1, \ldots, k_l$, whereas each sensor node N_i stores a subset $\mathcal{X}_i$ of the master keys, a key ring. Each sensor node N_i obtains additionally a set of keys $\mathcal{K}^i$ derived from the master keys. This is done with a pseudo random function as described in Section 4.1.2. Also the OBU obtains a set of keys $\mathcal{K}^{ID}$ derived from the TC's master keys. To complete the key setting of the use case 1 note that RANBAR does not require the storage of any keys whereas SANE assumes the initial storage of a seed $p_i^{(0)}$ per sensor node N_i based on a public generator function G for $p_i^{(0)}$ and subsequent seed values.

For the key setting of a *chargeable premium service* in the accident prevention scenario, one would further need to provide a real-time responsive access control. Here, we propose to use the same key sets as for the authentication in the use case 1. In case of using CDA, we choose a CDA derivate based on pairwise symmetric keys between each sensor node and the sink (respectively vehicle) similar to [7]. Each sensor node N_i stores a secret key $\hat{K}_{C(i)}$ that is unique for the cluster $C(i)$ the node N_i belongs to. This key will aim as a master key to derive the actual round key that will be used to encrypt the messages. Each OBU stores the master keys of each cluster.

Finally, for the complementary security architecture supporting a *post-accident investigation*, the proposed security modules require the storage of a public key K_q at each sensor node N_i and its corresponding private key K_p on a reader device *RD* of the forensic team. This key pair is used for tinyPEDS as well as for the access control scheme authorizing queries from the RD to the WSN.

Summing it up: For the full-fledged WSN roadside architecture, each sensor node N_i needs to store the tuple $(\mathcal{K}^i, \mathcal{X}_i, G, p_i^{(0)}, \hat{K}_{C(i)}, K_q)$, each vehicle's OBU stores the tuple $(\mathcal{K}^{ID}, \{\hat{K}_C\}_C)$, and each reader device *RD* of the forensic team purely stores K_p. The toll collect station *TC* stores the set of master keys $\mathcal{K}$.

The above described key setting for the proposed security architecture for WSN roadside islands has two benefits: Firstly, it takes into account the absence of any tamper-resistant modules within the WSN. Only public keys or secret keys which do reveal only some partial information in case of physical node capture are stored within the WSN island. Secondly, the key management between WSN islands and vehicles is as loose as possible. For use case 1, an OBU only needs to know the key set $\mathcal{K}^{ID}$. For use case 2, in addition, the keys for the clusters need to be stored at the OBU.

4. PROTOCOL SUITE

First, we describe the individual security building blocks which we use before we subsequently describe the full architecture protecting the introduced WSN services.

4.1 Security Building Blocks

4.1.1 Resilient Data Aggregation

RANBAR provides a general algorithm to resiliently aggregate data values. We will use the concrete algorithm for average computation that was introduced in the original paper by Buttyan et al. [5].

The algorithm receives as input a set X of sensor data readings. At first, the algorithm chooses a minimal subset of the data which is necessary to estimate the statistic distribution of all sensor values. For the average computation we assume a normal distribution on the sensor data. Then, the parameters of the distribution, the average and the variance, can be estimated from two values. RANBAR continues to check whether a set $\bar{X} \subset X$ of size larger than a given minimum size q can be found, that is close to the estimated distribution. This is done by computing the histogram of the full set X and successively removing one of the elements that contribute most to the distance between the histogram and the estimated distribution until the distance is smaller than a threshold δ.

If the size of the remaining set $\bar{X}$ is at least the required minimum size q, the average of the elements in $\bar{X}$ constitutes the aggregated result avg. Otherwise, in case the remaining set is smaller, the algorithm starts anew, choosing two random elements to estimate a distribution, and searching for a set $\bar{X}$. Once a defined maximum number of repetitions is reached without success, the algorithm aborts, assuming that too many sensors are malicious.

4.1.2 Key Derivation from a Master Key

We use a pseudo random function to derive multiple keys from a single master key. In the envisioned scenario, the

master key is generated at the TC by the administrator of the network. Given a master key k, k^{ID} denotes the key that is derived, using the string ID, i.e. $k^{ID} = \mathrm{PRF}(k, ID)$ for a pseudo random function PRF. If $\mathcal{K}$ is a set of keys $k_1, \ldots, k_l$, then $\mathcal{K}^{ID}$ denotes the set

$$\{\mathrm{PRF}(k_1, ID), \ldots, \mathrm{PRF}(k_l, ID)\}.$$

Only the derived keys $\mathcal{K}^{ID}$ are distributed to the vehicle passing the toll collect station TC.

To issue keys that are only valid until a time T, T will be included in the key derivation, i.e. $k_i^{ID,T} = \mathrm{PRF}(k_i, ID, T)$.

4.1.3 Multicast Authentication

The multicast authentication by Canetti et al. [6] was adapted to WSNs by Benenson et al. [4]. Parameters ω, that describes the number of node corruptions that should be tolerated, and a maximum attack probability q are chosen. The WSN operator chooses a key pool $\mathcal{K} = \{k_1, \ldots, k_l\}$ of l random keys. Every sensor node gets a subset $\mathcal{X}_i$ of $v = l/(\omega + 1)$ keys. We use a pseudo random function evaluated at i to determine the keys for N_i.

This can be combined with the key derivation of Section 4.1.2 to allow for multiple senders (see [6]). In our scenario, a OBU holds the key set $\mathcal{K}^{ID} = \{k_1^{ID}, \ldots, k_l^{ID}\}$. To authenticate a message m to the WSN, the OBU computes a message authentication code $\mathrm{MAC}_{k_i^{ID}}(m)$ with length of only one bit for each key $k_i^{ID} \in \mathcal{K}^{ID}$. We denote the string composed of the l 1-bit MACs for a query to the WSN m with $\mathrm{MAuth}_{\mathcal{K}^{ID}}(m)$. A node holding $\mathcal{X}_i$ can now compute the keys k_i^{ID} for the keys $k_i \in \mathcal{X}_i$ and verify the authentication at the respective positions.

4.1.4 Non-Manipulable Node Election

We apply a SANE [19] derivate based on predetermined random values to randomly elect a sensor node from a cluster to become a witness node. Besides reducing the communication overhead, the result of SANE is *predictable* if the seeds are known. This will be needed for the verification of the convergecast authentication.

We use a pseudo random function G, which generates a sequence of pseudorandom values of arbitrary length depending on a seed p. In other words once G is initialized with a seed p, it generates values $p^{(0)}, p^{(1)}, p^{(2)}, \ldots, p^{(n)}$.

The scheme works as follows: 1) Prior to deployment all nodes in a cluster C of a WSN roadside agree on G. 2) In the *commitment* phase, each sensor node N_i randomly chooses a seed p_i. Subsequently, node N_i broadcasts p_i in its cluster, so that each sensor knows the seeds of all nodes in its sector. This fixes each node's list of pseudorandom values $p_i^{(0)}, p_i^{(1)}, \ldots$. 3) In round t, at each node N_i, $|C|$ nodes announce their availability and their values $p_1^{(t)}, \ldots, p_{|C|}^{(t)}$ are treated as their random values. As the seeds and G are known, the random values of all nodes can be computed by each sensor node independently.

The mapping function for converting the random aggregate value R_i to a node ID is defined as follows: 1) Each node N_i stores the node IDs of the nodes in the set C in an ordered set L, such that L^0 is the lowest ID node and $L^{|C|-1}$ is the highest; 2) Each node N_i elects its witness as $N_w = L^{R_i \mod |C|}$.

Even if an attacker knows all the seeds that are chosen by the honest nodes *before* he has to reveal his seed, finding a value which suits his goals would be practically infeasible as she would have to pick one round for the attack in advance. The reason is that his initial choice determines his contribution for each round without the possibility to influence it afterwards.

4.1.5 Convergecast Authentication

We describe a new protocol for authenticating messages that the WSN returns to the querier. This scheme builds on the same keys as the multicast authentication for multiple senders described in 4.1.2.

A sensor node N_i authenticates a message with its own subset $\mathcal{X}_i$ of keys. If multiple nodes authenticate the message in this way, the message carries the authentication for a larger subset of keys of $\mathcal{K}$ than a single node holds. With this scheme the querier can get assurance that the response message is confirmed by multiple nodes, in this scenario by the aggregator and the witness node. Due to the predictability of SANE and deterministic key distribution, the OBU can verify that the authentication is computed by the actual aggregator and witness of this time interval. We use this kind of authentication for the accident prevention service and denote the procedure to compute the authenticator for message m with the key set $\mathcal{X}_i$ by $\mathrm{RAuth}_{\mathcal{X}_i}(m)$.

The choice of the keys by a pseudo random function allows the receiver to verify the identity of the sender.

4.1.6 Concealed Data Aggregation

To protect chargeable WSN premium services from unauthorized vehicles, we apply the concept of *concealed data aggregation* (CDA) originally proposed in [12]. CDA applies additive homomorphic encryption transformation to WSNs. Let an encryption transformation be $E : K \times Q \to R$ and the corresponding decryption function be $D : K \times R \to Q$. Given $a_1, a_2 \in Q$ and $k \in K$, a symmetric homomorphic encryption scheme provides

$$a_1 + a_2 = D_k\big(E_k(a_1) \oplus E_k(a_2)\big).$$

The symbol "+" represents an additively operation on words from the plaintext alphabet and the symbol "$\oplus$" represents the corresponding additive operation on words from the ciphertext alphabet.

To support *different* symmetric keys per encrypting party several approaches have been proposed: Castelluccia et al. [7] proposed a provably secure CDA derivate based on a streamcipher with the drawback that key-IDs of all involved nodes have to be transmitted during each aggregation process. Önen and Molva [16] use the CTR-mode encryption for homomorphic encryption in WSNs. Armknecht et al. [3] introduce a bihomomorphic encryption function to reduce the overhead for the key management in trade-off for a lower security level.

For the usage in a WSN roadside setting with communication between a WSN and a vehicular's OBU we use the first approach, though, we use cluster-wise keys to reduce the overhead in transmission. A concrete homomorphic encryption scheme is as follows: A value M is chosen as a system-wide parameter limiting the message space. A message $m \in [0, M-1]$ is encrypted with the key $\hat{k}_1^t$ as $c = E_{\hat{k}_1^t}(m) = m + \hat{k}_1^t \bmod M$. The key $\hat{k}_1^t$ is hereby constructed from the cluster's master key $\hat{K}_1$ with a pseudo random function for the time interval t. The sum of two

ciphertexts c_1 and c_2 computed with keys $\hat{k}_1^t$ and $\hat{k}_1^t$, respectively, can be decrypted with the key $\hat{k}_1^t + \hat{k}_2^t$.

At the end of an aggregation process, the vehicle receives the encryption of the aggregated sensed values which corresponds to a ciphertext encrypted with the aggregation of the keys of the responding clusters. The keys of all clusters have to be present at the OBU for decryption. However, the keys of clusters that could not contribute, e.g. due to exhaustion or unreliable channels, have to be omitted in the decryption process. Thus, the identifiers of those clusters have to be transmitted in addition.

4.1.7 Persistent Encrypted Data Storage

We recommend to establish a high security level for the long-term storage of aggregated sensed values within the WSN. We apply the *tiny persistent encrypted data storage* (tinyPEDS) [11] middleware for encrypted and aggregated storage of sensed environmental data over the time and/or over the region. For encryption within the roadside WSN we recommend to apply an *asymmetric* homomorphic encryption scheme. Each node of the WSN encrypts with a public key whereas only the reader device RD of the forensic team can perform decryption with the private key.

Environmental data representing e.g. the road condition monitored in the past can be stored in a ciphered way and still be aggregated (summed up) over time periods.

To deal with failing roadside sensor nodes, replica of aggregated encrypted values are transmitted at the end of each epoch to a neighboring aggregator node. Together with a query language and a suitable controlled flooding mechanism [11] this ensures that the forensic team can still read out data of the past even in case a fraction of the WSN is already exhausted.

A promising *asymmetric* privacy homomorphic candidate for the requirements of a WSN roadside is the ElGamal public-key encryption scheme on an elliptic curve $E(F_p)$ over a finite field F_p. The EC-ElGamal encryption scheme is based on the ECDLP. We apply

$$\begin{aligned} M &= map(a) \\ E_{K_q}(M;k) &= (R,S) \quad \text{where} \quad R = kG, S = M + kY \end{aligned}$$

with the public key $K_q = (E, p, G, Y)$, where $G \in E(F_p)$ is a generator point and $Y = xG$ for a random number $x \in \mathbf{Z}_{\#E(F_p)}$. The value $k \in \mathbf{Z}_{\#E(F_p)}$ is chosen at random for the encryption.

The function $map()$ is a deterministic mapping function used to map plaintext values a into "plaintext" curve points M and vice versa such that $map(a_1 + a_2) = map(a_1) + map(a_2)$. Decryption subsequently applies the reverse mapping function $rmap()$

$$\begin{aligned} D_{K_p}(E_{K_q}(M)) &= -xR + S = -xkG + M + xkG \\ a &= rmap(M) \end{aligned}$$

with the private key $K_p = x$.

As a homomorphic mapping function we use $map(a) = aG$. Note that solving $rmap()$ is equivalent to solving the DLP over an elliptic curve, which surely represents a computational drawback. The reader device of the forensic team must thus be powerful enough to solve $rmap()$ using a brute force approach. Solving the $rmap()$ for a one to two bytes plaintext on a desktop type RD would be in the range of up to 2.4 sec.

Depending on the number of precomputed points execution times on an 8-bit processor for an EC-ELGamal encryption vary between 1.19 sec to 2.48 sec [22] translating into 27.32 mJ to 49.24 mJ of energy consumption. Note that we apply the concept of *point compression* to reduce the resulting size of a cipher from $2|key|$ to $|key| + 1$. With a 160 bit key size together with header and meta data 46 bytes need to be transmitted over the RF IEEE 802.15.4 either for the storage of ciphered replica or for a query response to the forensic team. For comparison, tinyPEDS without encryption would cause a packet size of 18 bytes.

4.2 Protocol Composition for Accident Prevention

We assume a network that consists of multiple clusters and which consists of at least two aggregation levels. In every cluster, one node acts as aggregator node N_a and one node as witness node N_w, both determined by SANE. We describe the protocol for one witness node for each aggregator, however, the protocol can easily be generalized for multiple witness nodes. To achieve resilience, the messages between nodes have to be authenticated. We use the multicast scheme outlined in Section 4.1.3 for authentication in a way that enables all nodes to authenticate messages for other nodes. In this case, the WSN administrator holds the master-key set $\mathcal{K}$ of size l and every node holds v of these keys. In addition every node N_i holds the set $\mathcal{K}^i$ with l keys constructed as proposed in Section 4.1.2.

We describe the protocol in a cluster of the first (lowest) level, where the sensor data is aggregated. The protocol starts with a OBU broadcasting a value ID to initiate the aggregation protocol. The sensor nodes N_i transmit their data values x_i and a current timestamp t_i authenticated with their own key set $\mathcal{K}^i$ to their respective aggregator N_{a_1} and witness N_{w_1}. The aggregator node N_{a_1} checks the authentication and time of each result, and computes $avg = \mathrm{RANBAR}(X)$ for the set $X \subset \{x_1, \ldots, x_n\}$ of correctly authenticated and recent data from the sensor nodes. The aggregator node N_{a_1} forwards the result avg together with its current time t_{a_1} to the witness node N_{w_1}. The witness checks if avg is close to its own evaluation of RANBAR on the input data X and if the aggregator's time t_{a_1} is close to its own time. In this case, the witness N_{w_1} agrees and authenticates the aggrageted value avg and the aggregator's timestamp t_{a_1} with its key set $\mathcal{K}^{w_1}$. The witness sends the result to the next higher level aggregator and witness nodes, N_{a_2}, N_{w_2}, respectively. Thus, the protocol looks as follows:

$OBU \rightarrow *:$	ID
$N_i \rightarrow N_{a_1}, N_{w_1}:$	$N_i, x_i, t_i, \mathrm{MAuth}_{\mathcal{K}^i}(x_i, t_i)$
$N_{a_1}:$	compute $avg = \mathrm{RANBAR}(X)$
$N_{a_1} \rightarrow N_{w_1}:$	$N_{a_1}, avg, t_{a_1}, \mathrm{MAuth}_{\mathcal{K}^{a_1}}(avg, t_{a_1})$
$N_{w_1}:$	check if $\lvert avg - \mathrm{RANBAR}(X)\rvert < max$
$N_{a_1} \rightarrow N_{a_2}, N_{w_2}:$	$N_{a_1}, avg, t_{a_1}, \mathrm{MAuth}_{\mathcal{K}^{a_1}}(avg, t_{a_1})$
$N_{w_1} \rightarrow N_{a_2}, N_{w_2}:$	$N_{w_1}, \mathrm{MAuth}_{\mathcal{K}^{w_1}}(avg, t_{a_1})$

On higher levels, the aggregator computes $avg = \mathrm{AGG}(x_1, \ldots, x_n)$ for the already aggregated values x_i of the aggregators in the lower level. The function AGG can be addition or average computation. As long as the aggregator and witness nodes are not on the highest level, they authenticate the data using the function MAuth with their derived key set and send them to the aggregator and witness nodes on the next level as in the protocol above.

The aggregator node and the witness node on the highest level need to know the OBU's identity ID. Those nodes compute the authentication using the function RAuth. As a key, they compute a subset of the OBU's keys, using their key ring $\mathcal{X}_i$ of master keys. We denote these keys by $\mathcal{X}_i^{ID} = \{PRF(k, ID)\}_{k \in \mathcal{X}_i}$. The following overview shows the messages that are exchanged between the aggregator node N_{a_ℓ}, witness node N_{w_ℓ} of the highest aggregation level ℓ and the querier OBU:

$$\begin{array}{ll} N_{a_\ell}: & \text{compute } avg = \mathrm{AGG}(X) \\ N_{a_\ell} \rightarrow N_{w_\ell}: & N_{a_\ell}, avg, t_{a_\ell}, \mathrm{MAuth}_{\mathcal{K}^{a_\ell}}(avg, t_{a_\ell}) \\ N_{w_1}: & \text{check if } |avg - \mathrm{AGG}(X)| < max \\ N_{w_\ell} \rightarrow N_{a_\ell}: & N_{w_\ell}, avg, t_{a_\ell}, \mathrm{RAuth}_{\mathcal{X}_{w_\ell}^{ID}}(avg, t_{a_\ell}) \\ N_{a_\ell} \rightarrow OBU: & N_{a_\ell}, avg, t_{a_\ell}, \mathrm{RAuth}_{\mathcal{X}_{a_\ell}^{ID}}(avg, t_{a_\ell}), \\ & \mathrm{RAuth}_{\mathcal{X}_{w_\ell}^{ID}}(avg, t_{a_\ell}) \end{array}$$

The querier verifies the authenticity of the answer on the keys $\mathcal{X}_{w_\ell}^{ID} \cup \mathcal{X}_{a_\ell}^{ID} \subset \mathcal{K}^{ID}$ that are part of its keys obtained from the TC.

4.3 Protocol Composition for Premium Services

4.3.1 Access Control

If premium services are offered, the security architecture includes an access control to restrict the access to paying customers. The access control we present grants access based on the authenticity of queries sent to the WSN. A light-weight access control suffices because the number of data objects, classification levels, and available services will be small. We aim at granting user access to the service for a certain time rather than for a single access. The access control uses the multicast scheme of Section 4.1.3 in combination with the key derivation of Section 4.1.2. With these schemes, a set of personalised keys can be derived for a OBU. These keys allow the user to form authenticated queries to the WSN.

For the multicast authentication scheme, the administrator holds a set $\mathcal{K} = \{k_1, \ldots, k_l\}$ of l master keys and every node N_i holds a subset $\mathcal{X}_i$ of v of these keys. Every querying vehicle has an identity ID that identifies the OBU. In the following we assume that an OBU obtains access to the WSN for a certain time period. Other restrictions of the access are possible in a very similar way. When passing the TC, the vehicular driver can buy the access rights to the WSN. The access rights consist of a set $\mathcal{K}^{ID}$ of keys that are derived from the master keys using the identifier ID and a validity period T. We do not detail further on the protocol of purchasing the access rights at the TC.

We now describe the access of an OBU with identity ID that intends to query q at time t. Therefore, the OBU authenticates the query q together with the time t with its key set $\mathcal{K}^{ID}$ and sends the tuple

$$OBU \rightarrow *: \quad ID, T, q, t, \mathrm{MAuth}_{\mathcal{K}^{ID}}(q, t)$$

as her ticket to the roadside WSN. This message replaces the first message in the protocol for the public accident prevention service and propagates through the WSN. Each receiving node N_i checks the authenticity of the query as follows: At first N_i checks whether $t < T$, i.e. the user's key is still valid, and whether the OBU's query is recent, i.e. t is close enough to the node's time. N_i computes the OBU's keys k_i^{ID} for the keys $k_i \in R_i$ that N_i holds and checks the authentication for these keys. If one of the checks fails, N_i does not propagate the query and aborts the processing. Following this first message, the protocol proceeds to transmit and authenticated aggregate the data in analogy to the protocol for the free accident prevention service.

4.3.2 Concealed data aggregation

To protect the premium data from eavesdropping adversaries, we use CDA as described in Section 4.1.6. However, we adapt the scheme to the particular needs of the service. As resilience of data is as well important for the premium service and the nodes are exposed to a threat of being compromised, the sensor data is at first aggregated by the RANBAR algorithm. Thus, CDA is only used in a network with several clusters and at least two aggregation levels. Then the aggregator nodes on the first level encrypt the aggregated data and all upper aggregation nodes aggregate only ciphertexts. The keys are distributed such that all nodes in one cluster hold the same keys.

The first-level aggregation is done cluster-wise. The aggregator node and the witness node are both located in that cluster and hold an identical encryption key. Then aggregator and witness node compute at first $\mathrm{RANBAR}(X)$ as for the public service and will then compute and authenticate $E_K(avg), t_a$ with the cluster key K_C where C is the respective cluster. Having only one ciphertext per cluster helps also reducing the overhead of the encryption scheme. Now only the identifiers for each cluster has to be transmitted instead of identifiers of individual nodes.

To complement this service and to provide confidentiality on all levels, the data between the sensing nodes and the aggregator could be hop-to-hop encrypted, e.g. with the common cluster key K_C.

4.4 Protocol Composition for post-accident investigation

To make forensic analyses possible, the WSN stores regularly the sensor readings. This is done in an encrypted way by applying tinyPEDS. We extend tinyPEDS to work together with the witness-based approach that was introduced in this Section. This ensures that the storage of the data is as resilient as for the accident-prevention service. The process of storing the data within tinyPEDS can be activated once the WSN aggregates the data for the accident-prevention service, or can be executed independently.

We now describe the process from data transmission to data storage for the first aggregation level. The protocol for higher aggregation levels is analogous, with the only difference that the aggregation function AGG is applied to the received already aggregated input values.

Once the aggregator and witness nodes N_a and N_w receive the sensor data readings $X = \{x_1, \ldots, x_n\}$, they compute the average avg with the RANBAR algorithm. The difference to the accident-prevention is that this data is now stored with EC-ElGamal encrypted, instead of being transmitted to the next level. As the network knows only the public key K_q of the encryption scheme, the original data cannot be recovered inside the WSN. Though, due to the homomorphic property of ElGamal, the stored data can be aggregated further. The authenticiy of the data is ensured by incorporating tinyPEDS with the witness scheme. To enable nodes that have to aggregate encrypted data to check the authenticity of the stored data, it becomes necessary

to authenticate the ciphertexts instead of the data. This is done by the witness node N_w and the aggregator node N_a. Before the nodes authenticate the ciphertext, they need to agree on the validity of the ciphertext. This is done by sending the randomness r that is needed to compute the ciphertext to the witness, enabling the witness to encrypt the aggregated value *avg* in the same way. This value should be communicated encrypted, e.g. with the common cluster key. Finally, the aggregator node is responsible to store the data according to the tinyPEDS storage policy for data replica [11].

$$\begin{aligned} N_i \to N_a, N_w: &\quad N_i, x_i, t_i, \mathrm{MAuth}_{\mathcal{K}^i}(x_i, t_i) \\ N_a \to N_w: &\quad N_a, avg, t_a, r, \mathrm{MAuth}_{\mathcal{K}^a}(avg, t_a) \\ N_w \to N_a: &\quad N_w, \mathrm{RAuth}_{\mathcal{X}_w}(E_{K_q}(avg); r), t_a \\ N_a \text{ stores in tinyPEDS:} &\quad t_a, E_{K_q}(avg; r), \\ &\quad \mathrm{RAuth}_{\mathcal{X}_a}(E_{K_q}(avg; r), t_a), \\ &\quad \mathrm{RAuth}_{\mathcal{X}_w}(E_{K_q}(avg; r), t_a) \end{aligned}$$

To enable a forensic team with a reader device RD to carry out an investigation, the shared asymmetric key pair is used to authenticate the query

$$RD \to *: \quad t_{RD}, q, \mathrm{Sig}_{K_p}(t_{RD}, q).$$

The WSN will answer the query with the corresponding ElGamal encrypted data of tinyPEDS. Due to space limitations, for an insight into the query response algorithm we refer the reader to [11].

5. PERFORMANCE EVALUATION

5.1 Security Analysis

We firstly analyse the security of the protocol architecture and reduce attacks to the protocol to attacks on the applied components. Subsequently, we discuss the security of the components and point out reasonable parameter choices. For online attacks, where the adversary cannot predict if the attack will be successful, we consider an attack probability of 2^{-10} sufficient. In this case, the adversary needs to send on average 2^9 messages to succeed. In combination with an attack detection on the sensor nodes and a backoff timeout to restrict repetitive attempts, the adversary needs to be present for a considerable amount of time. This results in a denial of service attack that, however, is in principle possible with a locally present adversary jamming the wireless channel.

5.1.1 Security of the protocol architecture

For the *public safety service*, we aim at preventing the stealthy attack. We assume a successful attack, i.e. the OBU accepts a flawed value *avg* delivered in a message $N_{a_\ell}, avg, t_{a_\ell}, \mathrm{RAuth}_{\mathcal{X}^{ID}_{a_\ell}}(avg, t_{a_\ell}), \mathrm{RAuth}_{\mathcal{X}^{ID}_{w_\ell}}(avg, t_{a_\ell})$.

Then, the adversary has either 1) broken the authentication RAuth, 2) obtained the keys of N_{a_ℓ} and N_{w_ℓ}, i.e. broken the witness scheme, or 3) *avg* is poisoned from input of a lower level. For the latter case, the aggregator and witness nodes are the victims of a stealthy attack. A successful attack on this level reduces again to the cases 1), 2) and 3). On the lowest level, if the input to the aggregator is faulty such that it results in a faulty aggregation, this implicates that the assumptions of RANBAR do not hold.

For the *premium service*, in addition, the security goals confidentiality and access control are relevant. Confidentiality even against node compromise is guaranteed by the CDA approach. This service is implemented from the first aggregator level. The access control restricts access to OBUs that present a valid ticket $ID, T, q, t, \mathrm{MAuth}_{\mathcal{K}^{ID}}(q, t)$. A successful attack implies that a majority of nodes accepts this ticket as authentic. If the authentication MAuth is secure, the adversary was able to replay a valid ticket or obtain the keys to compute the authentication. For protection against replay attacks we use the timestamp t. We assume the keys are securely stored inside the OBU, such that the adversary cannot authenticate new messages by herself.

The service *post-accident investigation* implements public-key cryptography for authentication and data encryption. Similar to the access control for the premium service, a replay is prevented and a secure storage of the private key outside of the WSN in tamper-resistant RDs is assumed.

5.1.2 Security of the building blocks

The security of the *multicast authentication* MAuth depends on the total number of keys l, the key ring size v of the nodes and the length of the authenticator for each key. We assume the keys to have a length of at least 80 bit, such that offline key guessing is infeasible. The scheme is however only secure against small coalitions of corrupted nodes. An evaluation of the scheme is given in [6]. We assume as parameters $l = 80$ and $v = 12$ and use a one bit authenticator per key. If the adversary has compromised 5 nodes and knows their keys, the probability that an uncorrupted node will accept a message of the adversary is less than 0.07. We consider this reasonably small, as the adversary can only check the correctness of the authentication in an online attack and needs a considerable amount of nodes to accept the query to activate the service. One faked message will not affect the service owing to the resilience of RANBAR and the witness scheme (see analysis for the witness scheme).

This is also valid for the *convergecast authentication* algorithm RAuth and the *access control* which are based on the same key sets.

REMARK 1. *In a small network with n nodes, the total size of keys can be chosen as $l = n$ such that every node holds $v = 1$ individual key. In addition, every node N_i obtains the n keys $\{PRF(k_j, i)\}_{j=1\ldots n}$ derived for its identity. This is a special case that leads to unique shared keys between each pair of nodes. Certainly, the length of the authenticator per key has to be appropriately higher than one bit.*

The adversary can only send faulty information in the *witness-based scheme*, if the aggregator and all witness nodes are under her control. Let w denote the number of witness nodes for a given aggregator node. Then, if γ out of the N nodes are corrupted, the probability that those nodes cover the aggregator node and all witness nodes is $\binom{N-(w+1)}{\gamma-(w+1)} / \binom{N}{\gamma}$. If we assume a cluster size of 15 and allow for $\gamma = 5$ compromised nodes within the cluster, we need 4 witness nodes to drop below the attack probability of 2^{-10}. One witness per cluster, as we described the protocol in this paper, reduces the attack probabilty only to 0.1.

RANBAR is secure as long as enough nodes to reach the minimum size q for accepting are uncorrupted. A reasonable value for q would be half of the cluster size. In a cluster with, say, 15 nodes, $q = 8$ would allow for 7 compromised or crashed nodes.

The *concealed data aggregation* protects the data in transmission. The scheme we use is proven secure in [7]. We

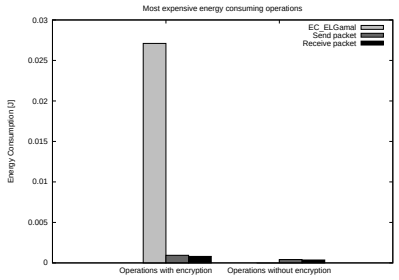

Figure 2: Energy consumption for elementary operations. Energy values for ECC-based encryption use 160 bit key and result in 46 byte ciphertext. The transmitted datagram is of size 53 byte.

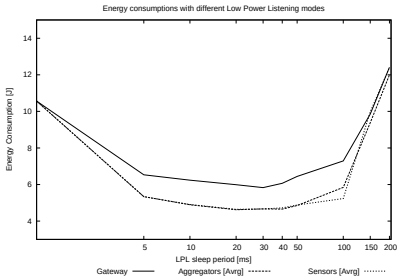

Figure 3: Energy consumption for the roadside WSN with respect to authenticated OBU queries every 7 sec and varying RI-sleeping intervals for 250s emulation time.

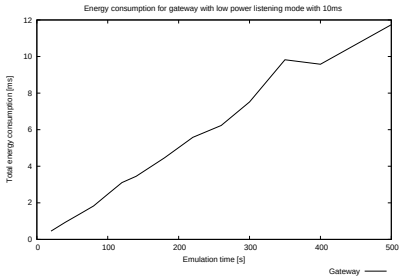

Figure 4: Total energy consumption for a gateway handling authenticated OBU queries every 7 sec with a 10ms RI-sleeping interval.

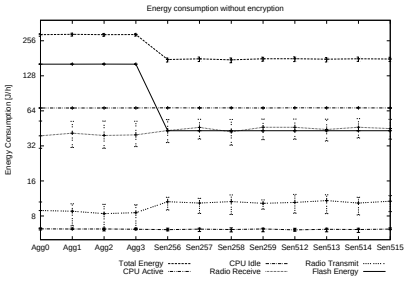

Figure 5: TinyPEDS energy consumption per hour without encryption for a reference roadside WSN with epoch:=1min, slot every 20 sec and idle time of 1000msec.

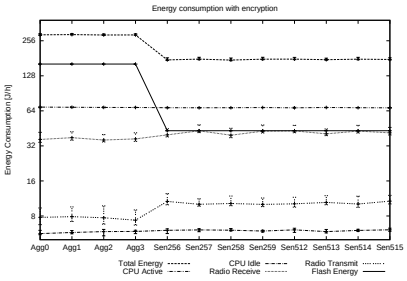

Figure 6: TinyPEDS energy consumption per hour with encryption for a reference roadside WSN with epoch:=1min, slot every 20 sec and idle time of 1000msec.

choose keys $\hat{K}_C$ of 80 bit length and a secure pseudo random function to generate the round keys.

5.2 Energy Analysis

For an accurate estimation of the energy consumption of the proposed WSN roadside architecture, we had to choose the RF configuration parameters, such that the configuration is energy efficient while ensuring a proper service for a passing vehicle. In particular we had to choose a sensor node's RI sleeping period. Assuming vehicles passing a WSN with 70km/h, and assuming a radio range of 30m to 50m we can infer that a vehicle stays 3-4 sec in the transmission range of the WSN. Since sending a 56 byte query from the OBU to the WSN takes 24 ms and transmitting convergecast traffic from the senor nodes via the aggregator nodes is a multitude of this period, the remaining time for the protocol components resilient data aggregation, multicast authentication, convergecast authentication and CDA should last no longer than 2.5-3.5 sec. For the AVRORA emulation, we set the RI's sleeping period of each sensor node to 10ms to evaluate use case 2 (access control)[2] and to one sec to evaluate use case 3 (tinyPEDS)[3].

Before we started simulating a WSN roadside running the complementary services access control and tinyPEDS, we evaluated elementary operations which majorly consume a sensor node's energy. Figure 2 shows that the ECC based encryption of one byte plaintext with a key size of 160 bits results in a 46 byte ciphertext and consumes 0.027 J. Transmitting and receiving this cipher results in a datagram of size 53 byte and consumes only a fraction of the above energy, namely $0.94 \cdot 10^{-3}$ J respectively $0.8 \cdot 10^{-3}$J. The corresponding values for the transmission and receiving of a datagram containing an aggregated plaintext value are $0.41 \cdot 10^{-3}$ J respectively $0.35 \cdot 10^{-3}$J. Please note, that, although not represented as an elementary operation, one should point out that read and write operations to persistent memory (after each epoch) result in similar energy consumption as public-key based encryptions. Energy consumption for the applied symmetric crypto schemes are negligible.

For the energy emulation of use case 3, we configured an epoch (EC-ElGamal encryption plus sending and storing of replica) with a duration of 1 min and one slot (sensing and sending from the sensor nodes to an aggregator node) every 20 sec. The envisioned roadside WSN consists of 12 Mica2 sensor nodes equipped with the RFM TR1000[4]. The WSN is subdivided into four clusters, each consisting of one aggregator node and two sensor nodes in a static setting.

For the energy emulation of use case 2 Figure 3 shows that the highest saving in energy consumption can be achieved by

[2]We do not consider the CDA application in use case 2.

[3]Obviously, for a simultaneous usage of use cases 2 and 3, we surely need to harmonize the sleeping period in a good balance for the conflicting real-time requirements.

[4]Although our demonstrator runs on MicaZ with RF IEEE 802.15.4 AVRORA only emulates Mica2 motes equipped with RFM TR1000.

setting the low power listening (LPL) mode with 20ms sleeping interval. However, the query response time increases with the increase in sleeping period, that is e.g. 0.989s and 1.135s for 10ms and 20ms, respectively. As shown in Figure 3 the most energy consuming node in the WSN is the gateway node. For estimating the lifetime of the roadside WSN we measured its energy consumption with RI sleeping period set to 10ms, see Figure 4.

For the energy emulation of the use case 3 we run ten times tinyPEDS each for 1h emulation in the modi a) with encryption (see Figure 6), b) without encryption (see Figure 5). Our observations are as follows:

Observation 1. Although the EC-ElGamal encryption is by far the most energy consuming elementary operation, for the overall energy consumption in the WSN the consumed energy is negligible. The consumption is dominated by the radio and by the persistent memory modules.

Observation 2. With the introduced parameter settings, the tinyPEDS roadside WSN lives for approximately four days.

Observation 3. Authenticated OBU queries with LPL mode of 10ms sleeping interval seem to be most promising as far as energy consumption and real-time responsiveness are considered. Compared to a fully active RI, the energy consumption is reduced by 41 %.

Observation 1 defends our choice to use asymmetric cryptography for non real-time responsive WSN applications. When moderately used asymmetric cryptography does not seriously effect the WSN's lifetime. However, the simultaneous running of protocol compositions for accident prevention and post-accident investigation also requires real-time responsiveness. Encryption durations in the range of 2.4 sec definitively require a careful implementation of tasks and events. Although the OS in use is tinyOS, the OS Contiki would be preferable here.

We derived Observation 2 by running tinyPEDS emulations for one hour. With an initial battery energy of $25 \cdot 10^3 J$ we can infer that an aggregator node exhausts in the range of three to four days, not considering the effect of flatly balancing the energy consumption due to aggregator node election. Obviously, the measured overall lifetime is by far not adequate for a WSN roadside solution. However, our feasibility study also shows that with a few adaptations of the configuration even for our reference platform it is possible to extend the lifetime to several months. By encrypting every epoch but persistently storing only, say, every hour, we can increase an aggregator's lifetime to approximately 180 days.

Observation 3 indicates that the energy consumption for authenticated queries and its propagation is uncritical. It takes approximately 0.989s and 0.621s for the query and response process from the vehicular's OBU to the WSN and vice versa. By sending queries every 7 sec, the gateway consumes 1380 mJ in 60sec simulation time. With an initial energ of two AA batteries it can handle approximately 185000 queries. without exhausting.

We believe that our energy measurements defend the principle feasibility to apply the proposed protocol suite for the protection of roadside WSNs. Obviously, for a real roll-out of WSN technology in the ITS sector one should also consider relying upon other forms of energy, e.g. renewable energy sources.

6. CONCLUSION

In this paper we proposed a protocol suite for a WSN roadside architecture protecting the complementary services accident prevention and post-accident investigation. Our evaluations show that the provided security level is appropriate against passive and active attacks as well as node capture. The energy analysis shows that although a single public key-based encryption operation consumes significant energy, it does effect the overall energy consumption of a roadside WSN only to a minor degree. Recently a demonstrator on the encrypted and persistent data storage in WSNs [13] has been presented. The integration into the ITS context has been demonstrated in October 2007 at the *ITS World Congress* in Beijing showing the security and reliability support for the complementary services post-accident investigation and accident prevention.

7. ACKNOWLEDGMENTS

The authors are indepted to Thomas Badura for his support in the performance evaluation. The authors are also most grateful to Roberto Baldessari and Andreas Festag who made the integrated ITS demonstrator possible. The work presented in this paper was supported in part by the European Commission within the STREP Project UbiSec&Sens of the EU Framework Programme 6 for Research and Development (FP6-2004-IST-4) (http://www.ist-ubisecsens.org). 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 or the European Commission.

8. REFERENCES

[1] F. Armknecht, A. Festag, D. Westhoff, and Z. Ke. Cross-layer privacy enhancement and non-repudiation in vehicular communication. KIVS WMAN, 2007.

[2] F. Armknecht, J. Girao, M. Stoecklin, and D. Westhoff. Re-visited: Denial of service resilient access control for wireless sensor networks. In *3rd European Workshop on Security and Privacy for Ad Hoc and Sensor Networks, ESAS 2006*, pages 18–31, 2006.

[3] F. Armknecht, D. Westhoff, A. Hessler, and J. Girao. A lifetime-optimized encryption scheme for sensor networks allowing in-network processing. *to appear in Elsevier Computer Communications*, 2007.

[4] Z. Benenson, L. Pimenidis, F. Freiling, and S. Lucks. Authenticated query flooding in sensor networks. In *4th IEEE Conference on Pervasive Computing and Communications Workshop*, pages 644–647, 2006.

[5] L. Buttyan, P. Schaffer, and I. Vajda. Ranbar: Ransac-based resilient aggregation in sensor networks. In *Security of Ad Hoc and Sensor Networks, ACM SASN 2006*, pages 83–90, 2006.

[6] R. Canetti, J. A. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast security: A taxonomy and some efficient constructions. In *INFOCOM '99*, pages 708–716. IEEE, 1999.

[7] C. Castelluccia, E. Mykletun, and G. Tsudik. Efficient aggregation of encrypted data in wireless sensor networks. In *2nd Annual International Conference on*

Mobile and Ubiquitous Systems: Networking and Services, pages 109–117, 2005.

[8] H. Chan, A. Perrig, and D. Song. Secure hierarchical in-network aggregation in sensor networks. In *In ACM Conference on Computer and Communication Security, CCS'06*, page 287, 2006.

[9] D. Dolev and A. Yao. On the security of public-key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[10] W. Du, J. Deng, Y. S. Han, and P. Varshney. A witness-based approach for data fusion assurance in wireless sensor networks. In *IEEE Global Communications Conference, GLOBECOM 2003*, pages 1435–1439, 2003.

[11] J. Girao, D. Westhoff, E. Mykletun, and T. Araki. TinyPEDS: Persistent encrypted data storage in asynchronous wireless sensor networks. *Elsevier Ad Hoc Journal*, 5(7):1073–1089, 2007.

[12] J. Girao, D. Westhoff, and M. Schneider. Cda: Concealed data aggregation in wireless sensor networks. in proceedings of the acm workshop on wireless security. In *ACM Workshop on Wireless Security, ACM WiSe'04*, 2004.

[13] A. Hessler, D. Westhoff, and E. Osipov. Encrypted persistent data storage for asynchronous wireless sensor networks (demo). 13th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'07), 2007.

[14] S. Lee, G. Pan, J. Park, M. Gerla, and S. Lu. Secure incentives for commercial ad dissemination in vehicular networks. In *The 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM MobiHoc 2007*, 2007.

[15] M. Manulis and J. Schwenk. Provably secure framework for information aggregation in sensor networks. In *The International Conference on Computational Science and its applications, ICCSA 2007*, pages 603–621, 2007.

[16] M. Önen and R. Molva. Secure data aggregation with multiple encryption. In *European Conference on Wireless Sensor Networks, EWSN'07*, 2007.

[17] E. Osipov. Tinylunar: One byte multihop communication through hybrid routing in wireless sensor networks. In *The 7th International Conference on Next Generation Teletraffic and Wired/Wireless advanced Networking, NEW2AN 2007*, 2007.

[18] M. Raya and J. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1):39–68, 2007.

[19] M. Sivrianosh, D. Westhoff, F. Armknecht, and J. Girao. Non-manipulable aggregator node election protocols for wireless sensor networks. In *5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, IEEE WiOpt 2007*, 2007.

[20] L. Thiem and K. Scholl. Verteiltes sniffen von IEEE 802.15.4 Netzen unter Zuhilfenahme eines WLAN Ad-hoc-Netzwerkes. 6th Fachgespräch drahtlose Sensornetze of GI/ITG, 2007.

[21] B. Titzer, D. K. Lee, and J. Palsberg. Avrora: Scalable sensor network simulation with precise timing. In *Proceedings of IPSN"05, Fourth International Conference on Information Processing in Sensor Networks, Los Angeles*, 2004.

[22] O. Ugus, D. Westhoff, R. Laue, A. Shoufan, and S. Huss. Optimized implementation of elliptic curve based additive homomorphic encryption for wireless sensor networks. In *2nd Workshop on Embedded Systems Security, WESS'2007*, 2007.

[23] F. Weingärtner and F. Kargl. A prototype study on hybrid sensor-vehicular networks. 6th Fachgespräch drahtlose Sensornetze of GI/ITG, 2007.

[24] D. Westhoff, J. Girao, and M. Acharya. Concealed data aggregation for reverse multicast traffic in wireless sensor networks: Encryption, key pre-distribution and routing. *IEEE Transactions on Mobile Computing*, 2006.