

# Encrypted Persistent Data Storage for Asynchronous Wireless Sensor Networks (Demo submission)

Alban Hessler  
NEC Europe Ltd.  
Kurfürsten-Anlage 36  
69115 Heidelberg, Germany  
+49 6221 43 42 1

alban.hessler@netlab.nec.de

Dirk Westhoff  
NEC Europe Ltd.  
Kurfürsten-Anlage 36  
69115 Heidelberg, Germany  
+49 6221 43 42 1

dirk.westhoff@netlab.nec.de

Evgeny Osipov  
Department of Computer Science and  
Electrical Engineering  
Luleå University of Technology  
97187 Luleå, Sweden  
+46 920 49 15 78

Evgeny.Osipov@ltu.se

## DEMO ABSTRACT

The demo presents the tiny persistent encrypted data storage (tinyPEDS). It is a distributed database for asynchronous wireless sensor networks that provides long term storage for measured data and ensures the confidentiality of the stored information. TinyPEDS is a complex middleware that integrates evolved network functionalities, advanced cryptographic transformations such as privacy homomorphisms, and user functionalities such as user-friendly queries support.

## 1. DEMONSTRATED RESEARCH CONTRIBUTIONS

The tinyPEDS middleware demonstrates the following research contributions. Firstly, we show a distributed data storage middleware that provides confidentiality and robustness, while being energy- and storage- efficient through aggregation. Secondly, we present the implementation of an efficient asymmetric privacy homomorphism based on elliptic curve cryptography. Finally, we demonstrate tinyLUNAR, a reactive ad-hoc routing protocol adapted for wireless sensor networks. The following sections describe each part in more detail.

### 1.1 TinyPEDS

TinyPEDS [1] is a middleware that offers a secure long term logging of the collected environmental data over time and over some regions. It targets applications where the connection to a sink device cannot be insured at all time. The reasons for this maybe manifold: deployment in remote regions, sink node failure, or attacks, to name only a few. In many scenarios indeed, the cost of a base station can be too high. Hence, accessing to the logging at a later time directly from the WSN is an effective alternative to using a sink.

TinyPEDS operates in a wireless sensor network (WSN) illustrated in Figure 1 formed by  $N$  sensor nodes arbitrarily

located in a monitored area. The area is divided into regions and each region is assigned an identifier. The

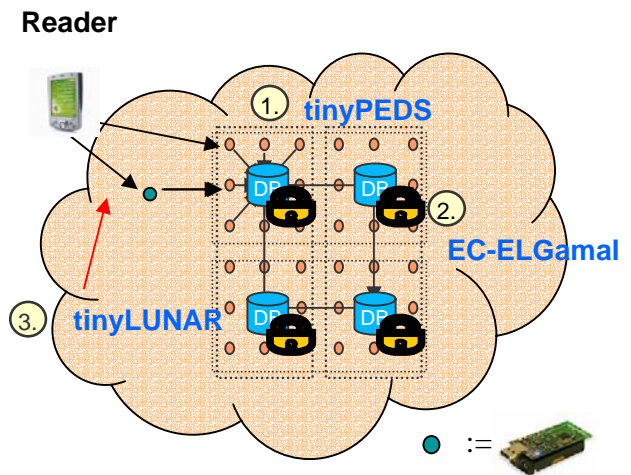


Fig. 1 TinyPEDS: Main components.

identifiers organized in a clockwise fashion.

The network is logically divided into clusters, where in each cluster at least one node is elected as a cluster head. We will refer the cluster head node in this region as to the *primary aggregator node*. Correspondingly we will refer the cluster head nodes in the neighboring regions as to *backup aggregator nodes*. The *reader* is a user operated device, which collects and processes the measured data from the sensor network. The reader allows a user to form location oriented data queries using a high level SQL-like language.

TinyPEDS includes the following features:

- user-friendly front end with graphical and SQL-like queries,

- mapping from human-friendly areas and time to WSN clusters and system epochs,

term of speed. A paper describing the implementation is currently under submission [3].

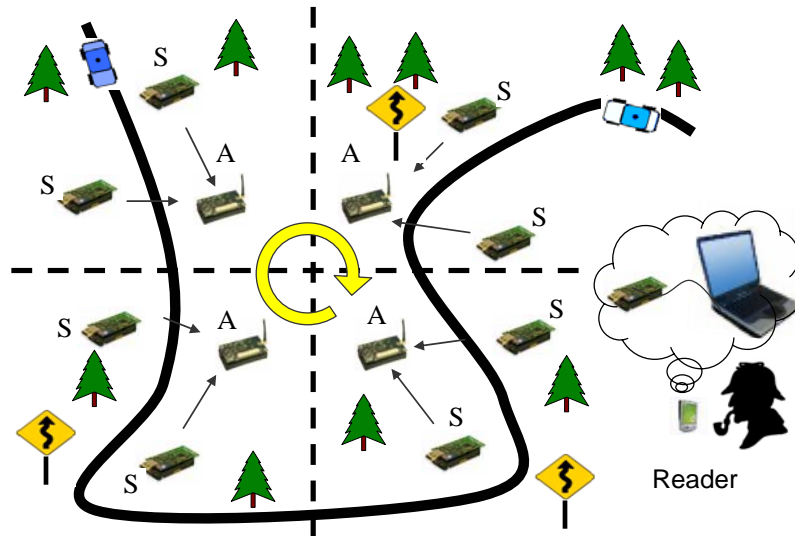


Fig. 2. Considered scenario and demonstration setup. Notations in the figure: S – sensor nodes (telosB), A – aggregator nodes (MicaZ), Reader is a laptop with a TelosB mote connected via serial interface. The routes between nodes are established by tinyLUNAR routing protocol.

- WSN query translation, dissemination and processing,
- hierarchical and thus scalable network topology,
- continuous collection and aggregation of data, which is then encrypted and saved on the mote memory,
- backup of the encrypted aggregate to neighboring nodes for robustness purposes,
- lifetime of the WSN is enhanced by a cluster head election algorithm (not implemented yet).

### 1.2 EC-ElGamal Privacy Homomorphism

The main primitive in tinyPEDS to protect the aggregated data is an asymmetric privacy homomorphism. It is instantiated by the EC-ElGamal encryption transformation. The choice was based on its small ciphertext size with an acceptable computation complexity while offering to perform additive operations on encrypted data [2]. Another advantage is its simple key distribution and its resistance to node compromises. An attacker cannot decipher any message, since he can only read out non-sensitive information, e.g. public key. Indeed the secret key is known by the incorruptible reader device only. Due to its storage optimization implementation, the code has a low memory footprint while performing very fast, making it suitable for WSNs. To our best knowledge, it provides the smallest program memory published so far and is second in

### 1.3 TinyLUNAR

The routes from sensor nodes to their primary cluster head as well as from the primary cluster head to the backup cluster head are established and maintained by tinyLUNAR. TinyLUNAR is the adopted to the specifics of sensor networks connection oriented routing scheme originally developed for mobile wireless ad hoc networks. The Lightweight UNDERlay Adhoc Routing (LUNAR) [4] is a layer 2 protocol that utilizes an extended label-switching forwarding technology. The major property of LUNAR is simplicity of implementation in comparison to other protocols developed for MANETs. This is achieved by reducing the route maintenance phase of the protocol to a minimum: In LUNAR all established path automatically and periodically expire and rebuild again upon demand from the application. TinyLUNAR inherits the simplicity of its predecessor. In addition it offers a flexible interface to the application level programmer to specify the destination node as a tuple of location and role parameters, which allows specification of the destination node as “An aggregator node in cluster X”. The current implementation of tinyLUNAR in TinyOS 2.x for MICA and Telos motes offers a competitive performance and stability compared to its counterparts, e.g tinyAODV.

## 2. DEMO DESCRIPTION

Due to its modular architecture, tinyPEDS can be used in many situations. Modules can be replaced according to the utility, security and lifetime needs. For this demonstrator, we choose a vehicular scenario, as depicted in Figure 2, as one example of the applications that tinyPEDS can support. In this scenario, a WSN is deployed in an area with several dangerous curves on a remote alpine road. Because of local shadows, there can be very confined places where temperature is lower and where ice forms. The on-board sensors of the car are not suited to detect such danger in advance. Hence the WSN could warn drivers about the road condition. Concurrently, the WSN could store the local conditions on their local memories with the tinyPEDS middleware. In case of an accident, a forensic team could then determine the exact condition of the road at the time of the accident. This can be used as a proof later to measure the level of responsibility of the driver, which has consequences with regard to law enforcement and to the car insurance.

### 2.1.1 Demo execution

The demo runs two distinct processes: On one hand, once the service has started, the middleware collects, enciphers, stores and backups environmental data continuously. This can be observed at any time, without the user intervention. On the other hand, the demo allows the user to build his own query and send it to the WSN. He will be able to see the request and the encrypted response flowing through the network. Finally, the result of the query is deciphered and revealed to the user on a GUI.

### 2.1.2 Requirements

The demonstrator needs a minimum of a dozen sensor motes of the class of the TelosB or MicaZ. Moreover, two motes are needed for the Reader device. One acts as a sniffer and is used to display packets transmitted in the network. The other one is the Reader, which sends and receives queries to the WSN. Both motes are connected to a laptop, which will display the front-end of the demo. All these equipments will be brought by us.

### 2.1.3 Additional requirements

Power access is necessary for the laptop, the beamer and possibly the programming board of the MicaZ sensor motes. I think that these 3 devices work as well for 110V as for 220V grids.

The demonstrator needs a horizontal area of about 2 x 2 meters to deploy the motes. A vertical area would be needed for a poster/beamer presentation.

The setup time is rather short. If there is no incident (mote failure, interferences, or so), the demo can be setup in 10 minutes.

### 2.1.4 Wireless access

There is no need for WLAN or Internet access. The motes communicate in the range of 2.4 GHz, therefore other demos in the same frequency range or a very dense WLAN network could interfere with the tinyPEDS demo.

### 2.1.5 Student awards

The demo is not eligible for student awards.

### 2.1.6 Acknowledgement

The work described in this submission is based on results of IST FP6 project UbiSec&Sens. UbiSec&Sens receives research funding from the European Community's Sixth Framework Program. Apart from this, the European Commission has no responsibility for the content of this paper.

## 3. REFERENCES

- [1] J. Girao, D. Westhoff, E. Mykletun and T. Araki, 'TinyPEDS: Tiny persistent encrypted data storage in asynchronous wireless sensor networks', *Elsevier Journal on Ad Hoc Networks*, 2007
- [2] E. Mykletun, J. Girao and D. Westhoff, Public Key Based Cryptoschemes for Data Concealment in Wireless Sensor Networks, in '*IEEE International Conference on Communications*', 2006
- [3] O. Ugus, A. Hessler and D. Westhoff, Performance of Additive Homomorphic EC-ElGamal Encryption for TinyPEDS, Under submission at 6. *GI/ITG KuVS Fachgespräch „Drahtlose Sensornetze“*, RWTH Aachen, 2007
- [4] C. Tschudin, R. Gold, O. Rensfelt, and O. Wibling, LUNAR: a lightweight underlay network ad-hoc routing protocol and implementation., in *Proc.NEW2AN'04*, St. Petersburg, Russia, Feb. 2004.