



BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
DEPARTMENT OF TELECOMMUNICATIONS

RESILIENT DATA AGGREGATION AND AGGREGATOR NODE ELECTION IN SENSOR NETWORKS

Collection of Ph.D. Theses
of
Péter Schaffer

Research Supervisor:
Levente Buttyán, Ph.D.



Budapest, Hungary

2009

1 Introduction

Sensor networks are distributed systems, consisting of hundreds or thousands of tiny, low-cost, low-power sensor nodes and one or few powerful base stations. Typically, sensors measure some physical phenomena and send their measurements to the base station using wireless communications. The base station performs data processing functions and provides gateway services to other networks (e.g., the Internet). Sensor nodes are able to transmit messages only within a short communication range, therefore, it is envisioned that the sensors form a multi-hop network in which the nodes forward messages on behalf of other nodes towards the base station and back to the nodes from the base station. In order to reduce the total number of messages sent by the sensors, in-network processing may be employed, whereby some sensor nodes perform data aggregation functions. Aggregator nodes collect data from surrounding sensors, process the collected data locally, and transmit only a single, aggregated message towards the base station.

As sensor networks may perform very important tasks (e.g., in homeland security or military applications), it is reasonable to assume that these networks will face attacks of various type. Even if we assume that the adversary is not powerful or that the nodes are tamper resistant, the adversary can still perform *input based attacks*, meaning that it can directly manipulate the physical environment monitored by some of the sensors, and in this way, it can distort their measurements and the output of the aggregation mechanism at the base station. This can be achieved by an attacker by altering the environmental parameters around some sensors and corrupt their readings. This type of attack cannot be detected, nor prevented, by cryptographic means. In addition, this type of attack is relatively easy to carry out: Firstly, an attacker can easily approach a sensor node, as sensor networks are typically assumed to operate in an unattended manner. Secondly, corrupting the measurement of a nearby sensor does not require sophisticated mechanisms, but in most of the cases, everyday tools can be used effectively (e.g., a lighter, a pocket lamp, or a glass of water can be used to corrupt temperature, light, and humidity measurements, respectively). Unfortunately, many useful aggregation functions are sensitive to even a single compromised sensor reading, meaning that their output can be arbitrarily modified by appropriately modifying a single sensor reading. Depending on the nature of the application, this may have fatal consequences.

There is already a well-studied part of statistics that deals with observations that deviate from the pattern set by the majority of the data; this field is called *robust statistics*. However, robust statistical tools may be inappropriate for sensor network applications. Generally, robust and resistant methods can only detect certain configurations of deviating observations, and the ability to detect such observations rapidly decreases as the sample size increases [Oli05]. In our scenario, an attacker is able to produce any kind of deviations (i.e., outliers) according to its will, thus, methods that are only able to detect certain configuration may not be satisfactory. Moreover, most robust and resistant methods have high computational complexity, which makes them unsuitable for low-power, low-end sensor nodes. Finally, resistant estimators may outperform classical estimators when deviating elements are present but they are far worse if no outliers are present [Oli05]. These problems motivated my research aiming at developing novel solutions for sensor networks that are able to cope with a determined attacker.

In the corresponding literature, the methods that aim at alleviating the problem of input based attacks in the context of sensor networks are called resilient data aggregation schemes (see e.g., [Wag04]). In order to circumvent the disadvantages of robust statistics, in Theses 1, I propose to apply two-phase solutions for resilient data aggregation that, in the first phase, analyze the sample and search for an attack using statistical hypothesis testing, and then, if no attack was detected, perform the aggregation in the usual way in the second phase. Otherwise, if an attack is detected, then special operation is performed in order to mitigate the effect of the attack on the aggregation result.

Then, in Theses 2, I propose a sample filtering approach, called RANBAR, for resilient aggregation that relies on the RANSAC (RANdom SAMple Consensus) paradigm. The RANSAC paradigm gives us a hint on how to instantiate a model if there are a lot of compromised data elements. The most interesting

property of the paradigm is that it suggests to use as few elements for instantiating the model as possible, on the contrary to classical statistical methods that usually suggest to use as many elements as possible.

Beside resilient data aggregation in sensor networks, in Theses 3, I deal with the problem of aggregator node election in sensor networks. The typical topology for a sensor network is a tree, where the base station is the root, and the non-leaf nodes are called aggregator nodes. As aggregator nodes perform more task than usual nodes (i.e., they also have to collect and aggregate measurements), these nodes use more resources than the regular sensor nodes. For this reason, it is desirable to change the aggregators from time to time, and thereby, to better balance the load on the sensor nodes.

In Theses 3, I propose an aggregator node election protocol for wireless sensor networks, called PANEL, which uses the geographical position information of the nodes to determine which of them should be the aggregators. PANEL ensures load balancing in the sense that each node is elected aggregator nearly equally frequently. Moreover, an important design criterion of PANEL was to perform aggregator node election in a non-manipulable way, i.e., to ensure that none of the nodes can become aggregator more frequently than the others. Otherwise, an attacker could force its compromised node to become aggregator all the time, and thus, it might manipulate the aggregated data collected from a larger set of common sensors continuously. Beside this non-manipulability property, PANEL also achieves a high level of security in the sense that it can defend against various attacks aiming at distorting the aggregation result, or ruining the aggregator node election process completely.

2 Research Objective

The goal of resilient data aggregation is to alleviate the problem of input based attacks (or alternatively called *environment altering attacks*). More specifically, resilient data aggregation schemes try to minimize the effect of an environment altering attacker at the output of the aggregation function. My goal is to understand the design principles of resilient data aggregation schemes, and based on this understanding, to develop novel resilient data aggregation algorithms that can be applied in sensor networks and that, at the same time, outperform resilient aggregation algorithms proposed so far in terms the distortion an attacker is able to cause.

Aggregator nodes aim at collecting the measurement data from other nodes and perform some kind of aggregation in order to reduce the amount of the information that has to be sent to the base station. As aggregator nodes perform more task than usual nodes, these nodes use more resources than the regular sensor nodes. My objective in this topic is to overcome the problem of this unbalanced energy consumption. My aim is to propose a new aggregator node election protocol that flatly balances the energy consumption of the network and outperforms existing aggregator node election protocols in this sense.

3 Methodology

My results related to resilient data aggregation in sensor networks rely heavily on probability theory and on statistics. I always consider the readings of the sensors as random variables, whether independent or correlated. This abstraction makes possible to me to use the notation and the knowledge of statistics. My results are primarily analytical, but in some cases, the complexity of the problem demanded the usage of simulation tools and empirical analysis. In the latter cases and in the case of numerical analyses, I use the Maple, Mathematica and Matlab programs.

My results and inferences related to aggregator node election in sensor networks are based on extensive simulations. I always assume some typical network topologies and performed simulations to see how the proposed protocols behave in terms of energy consumption and clustering capabilities. The performed simulations are usually comparative, i.e., I compare the proposed solution to another well-known algorithm. For these simulations, I used the Matlab, TOSSIM, and PowerTOSSIM programs.

4 New Results

4.1 The Sample Halving Approach to Resilient Data Aggregation

A potential problem in sensor network application scenarios is that sensor readings can be compromised before they reach the base station or the aggregator node. This can be achieved by an attacker for example by altering the environmental parameters around some sensors and thus corrupt their readings. This type of attack cannot be detected, nor prevented, by cryptographic means.

THESES 1: *I propose a two-sample homogeneity test, called the sample halving approach, for mitigating the environment altering attack. I present two usage mechanisms of this general approach; one for independent, and one for correlated sensor readings. [J1] [C1] [C4] [P2]*

The sample halving approach performs consistency checking of the sample by halving the sample and comparing the two halves in a statistical fashion (in other words, it performs statistical data splitting). Below, I present two slightly different usage mechanisms of this general idea. First, I consider the case when the received sample consists of independent elements, and then, I apply the sample halving approach to samples with correlated elements. In both cases, I separate attack detection from aggregation. The halving of the sample and the cross-checking is performed in the attack detection phase, and the aggregation (i.e., the calculation of the desired statistical functions) is performed after the attack detection phase has indicated that the sample is intact.

THESIS 1.1: *I propose a new model of resilient data aggregation in sensor networks assuming independent sensor readings. In this model, the aggregator analyzes the received sensor readings and tries to detect unexpected deviations before the aggregation function is called. The objective of the attacker in this model is to corrupt sensor readings in such a way that the distortion at the output of the aggregator is maximized, and at the same time, the attack remains undetected. [C4]*

My model of data aggregation with attack detection in the independent case is illustrated in Figure 1. I assume that there are n sensors, which perform some measurement and send their readings to a base station. The base station aggregates the received data; the objective of this aggregation is to estimate the value of an unknown parameter θ . I represent the reading of the i th sensor by a random variable X_i , whose distribution is a function of θ . For instance, θ may be the average temperature, and X_i 's distribution may be $\mathcal{N}(\theta, 1)$, the Gauss distribution with mean θ and variance 1. I assume that X_i ($i = 1, 2, \dots, n$) are identically distributed and independent. $\bar{X} = (X_1, X_2, \dots, X_n)$ is the vector that contains the readings of all sensors.

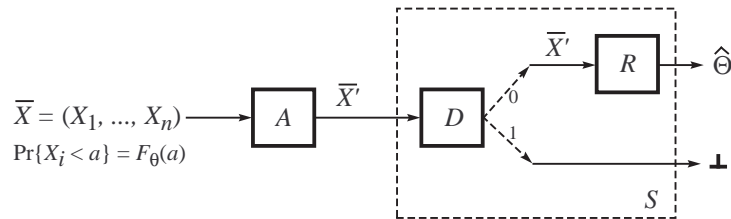


Figure 1: Model of data aggregation with attack detection

The adversary is allowed to modify the sensor readings before they are submitted to the aggregation function. This is modelled by a function A , which inputs the original sensor readings \bar{X} and outputs the modified vector \bar{X}' .

Formally, the operation of S is described as follows:

$$S(\bar{X}') = \begin{cases} R(\bar{X}') = \hat{\Theta} & \text{if } D(\bar{X}') = 0 \\ \perp & \text{if } D(\bar{X}') = 1 \end{cases} \quad (1)$$

where \perp is a special symbol that means that an attack was detected.

I assume that the adversary wants to maximize the distortion d of the aggregation function, which I define as follows:

$$d = \mathbb{E}[|\theta - \hat{\Theta}|] = \mathbb{E}[|\theta - R(A(\bar{X}))|] \quad (2)$$

In addition, I assume that the adversary wants to remain hidden, or more precisely, the adversary wants to keep the probability of successful detection of an attack under a given value p^* :

$$P\{D(\bar{X}') = 1\} = P\{D(A(\bar{X})) = 1\} \leq p^* \quad (3)$$

I assume that the adversary knows the detection algorithm D (including the a priori knowledge used in the algorithm) and the aggregation function R . I further characterize the adversary by the number $t < n$ of sensors that it has compromised. This means that \bar{X} and \bar{X}' differ in t positions.

The novelty of my model compared to [Wag04] is that I apply an attack detection step and, if there is no attack, I can use even aggregators that are considered to be not resilient against an attacker that can modify the measured parameters of the environment.

THESIS 1.2: *I propose a two-sample attack detection algorithm based on sample halving that fits the model described in Thesis 1.1 in case of a specific attacker. This specific attacker can modify the readings of a subset of the sensors selected before the attack, and the modification consists in adding a positive constant value to the reading of each selected sensor. [C4]*

I consider an adversary who can observe and modify the readings of $t \ll n$ sensors (selected before the attack). The adversary attacks by adding a constant value $m > 0$ to the reading of each selected sensor. I assume that the sensor readings X_i ($1 \leq i \leq n$) are independent and identically distributed. I assume that nothing is known about this distribution except for the fact that its variance is 1.

The attack detector uses the following algorithm. It first computes $Z_1 = X'_1 + \dots + X'_{n/2}$ and $Z_2 = X'_{n/2+1} + \dots + X'_n$, where, for simplicity, I assume that n is even, and then it computes $W = Z_1 - Z_2$. It is known from the central limit theorem that if there was no attack, then the distribution of W would be approximately $\mathcal{N}(0, \sqrt{n})$, the Gauss distribution with mean 0 and standard deviation \sqrt{n} . Therefore, it is suspicious if $|W|$ is not close to 0. The attack detection algorithm uses a threshold $h_\alpha > 0$ in the natural way:

$$D(\bar{X}') = \begin{cases} 1 & \text{if } |W| > h_\alpha \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The value of h_α is determined by a parameter α of the detection algorithm that represents the probability of false detection in the case when there is no attack (H_0 hypothesis):

$$P\{|W| > h_\alpha \mid H_0\} = 2 - 2 \cdot \Phi(h_\alpha/\sqrt{n}) = \alpha \quad (5)$$

The relationship of h_α and α is illustrated in Figure 2. As a matter of fact, this algorithm is highly related to the two-sample U-test, however, while this latter test requires normality, the idea of the sample halving approach can be considered as a more general one as it can be applied to any parameterized distribution (e.g., to asymmetric distributions as well).

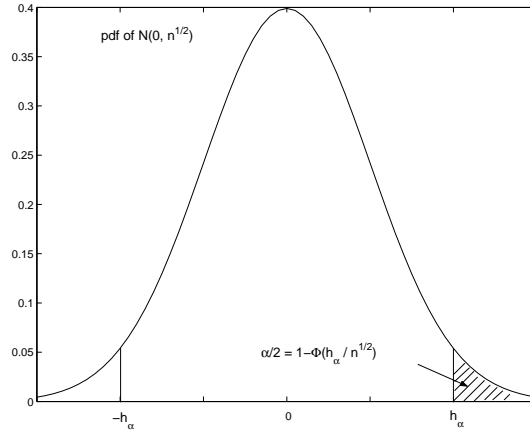


Figure 2: The value of h_α is determined by the probability α of false detection in the case when there is no attack, which corresponds to the tails of the distribution $\mathcal{N}(0, \sqrt{n})$

THEESIS 1.3: *I derive analytically the attack detection probability of the algorithm proposed in Thesis 1.2 in case of a specific attacker presented in Thesis 1.2. [C4]*

The expected value $\mathbb{E}[W]$ of W is a multiple of m , and it lies in the interval $[-tm, tm]$. Indeed, if t_1 denotes the number of compromised readings in the first half $X'_1, \dots, X'_{n/2}$ of the readings, and t_2 denotes the number of compromised readings in the second half $X'_{n/2+1}, \dots, X'_n$ of the readings, where $t_1 + t_2 = t$, then

$$\begin{aligned}
 \mathbb{E}[W] &= \mathbb{E}[X'_1 + \dots + X'_{n/2}] - \mathbb{E}[X'_{n/2+1} + \dots + X'_n] \\
 &= \left(\frac{n}{2} \cdot \theta + t_1 \cdot m\right) - \left(\frac{n}{2} \cdot \theta + t_2 \cdot m\right) \\
 &= (t_1 - t_2) \cdot m
 \end{aligned} \tag{6}$$

Therefore, we can write the following for the probability of detection in the case when there is an attack:

$$P\{D(\bar{X}') = 1 \mid H_1\} = \sum_{\ell=-t}^t P\{|W| > h_\alpha \mid \mathbb{E}[W] = \ell m\} \cdot P\{\mathbb{E}[W] = \ell m\} \tag{7}$$

Using some combinatorics, one can evaluate Equation (7) for given values of the parameters. Figure 3 illustrates the result of this calculation for $n = 100$ and $\alpha \approx 0.05$ (which gives $h_\alpha = 20$). The left subfigure corresponds to odd values of t , while the right subfigure corresponds to even values of t . The different curves belong to different values of t .

It is easy to see that if the adversary wants to keep the attack detection probability below a given threshold p^* , then the distortion that it can achieve is severely limited. For instance, if $p^* = 0.3$, then the distortion cannot be larger than 0.5 even if 9 out of 100 sensors are compromised. For the same value of p^* , the maximum achievable distortion reduces to about 0.1 if only 1 compromised sensor is used in the attack. Interestingly enough, the upper bound on the achievable distortion does not depend on the value of θ (i.e., the parameter to be estimated), which means that the relative distortion d/θ can be very small for large values of θ .

In the following, I present my work related to samples that are not independent, but correlated. Correlation among the sample elements is a naturally existing phenomena when considering measurement data, and thus, it should be considered when dealing with sample produced by sensor networks. For

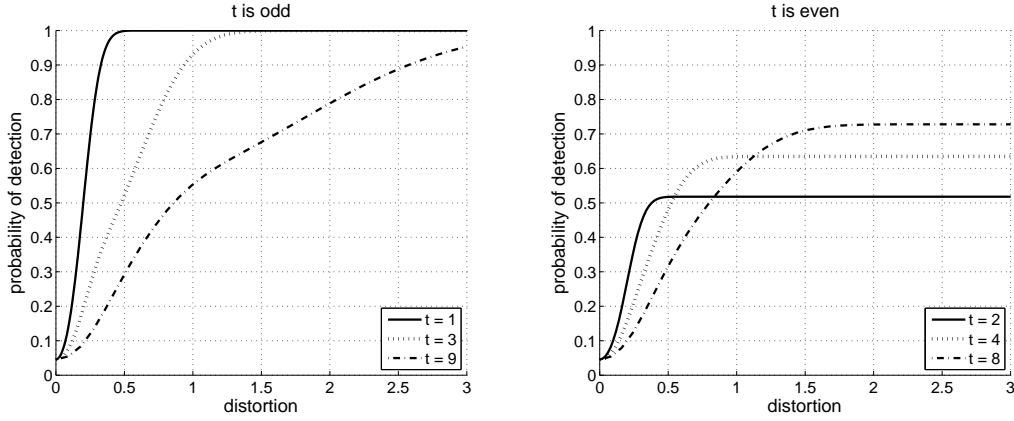


Figure 3: The attack detection probability as a function of the distortion achieved by the adversary for $n = 100$ and $\alpha \approx 0.05$ (which gives $h_\alpha = 20$).

my investigations, I use a straightforward sensor network data model which is able to keep track with correlation, namely, it deals with the pairwise correlation coefficient among the sample elements during the calculations.

THESIS 1.4: *I propose the Attack Detection Algorithm in conjunction with the Enhanced Data Aggregation Algorithm for resilient data aggregation that exploit linear correlation among the sample elements. [J1] [C1] [P2]*

The algorithm that is designed to detect the attack is Algorithm 1.

Algorithm 1 $Det(x_1, x_2)$ Attack Detection Algorithm

- 1: Randomly select one element from the sample $\{x_1, x_2\}$ and let the selected element be denoted by x' , the remaining one by x''
 - 2: Calculate the $(1 - \alpha)\%$ confidence interval for x'' conditioned on x' according to the p.d.f. $p_{X''|X'}(\cdot|x')$
 - 3: **if** x'' is inside this confidence interval **then**
 - 4: $D = 0$ (* no attack detected *)
 - 5: **else**
 - 6: $D = 1$ (* attack detected *)
 - 7: **end if**
-

This straightforward approach already exploits correlation by using the conditional probability density functions (p.d.f's) $p_{X_1|X_2}(\cdot|\cdot)$ and $p_{X_2|X_1}(\cdot|\cdot)$, which are assumed to be known. (X_i denotes the i th uncompromised sample element.) I note, however, that the knowledge of $p_{X_1|X_2}(\cdot|\cdot)$ and $p_{X_2|X_1}(\cdot|\cdot)$ does not imply the a priori knowledge of the p.d.f. of the measurement data at individual sensors. For example, a given conditional p.d.f. $p_{X_1|X_2}(\cdot|\cdot)$ gives a different joint p.d.f. for different distributions of X_2 , which then results in different marginal distributions for X_1 . Consequently, I do not assume any a priori knowledge about the expected value of the measurement data.

The output of Algorithm 1 can be applied in selecting the adequate way of data aggregation. My approach is formalized in the Enhanced Data Aggregation Algorithm (Algorithm 2), where output y is the aggregate of the input, while the output denoted by y_{extr} is the minimum distortion output when we do not use outlier filtering. y_{extr} is usually calculated as an extrapolation based on the output of the previous uncompromised outputs.

The output of the Enhanced Data Aggregation Algorithm is interpreted as the aggregate value of the

Algorithm 2 Enhanced Data Aggregation Algorithm

- 1: Take both of the readings and apply the Attack Detection Algorithm $Det(x_1, x_2)$
 - 2: **if** $Det(x_1, x_2)$ indicates an attack **then**
 - 3: Output = y_{extr}
 - 4: **else**
 - 5: Output = y
 - 6: **end if**
-

current round. Using the Attack Detection Algorithm and the Enhanced Data Aggregation Algorithm one can notably reduce the distortion of the aggregate compared to the case when aggregation is performed without prior analysis.

One can use the above algorithms for arbitrarily sized samples by halving the sample into two partitions and compressing the partitions into one element each. A small modification is needed in Algorithm 1, namely, one has to use the conditional p.d.f's of the averages conditioned on each other, as instead of $p_{X_1|X_2}(\cdot|\cdot)$ and $p_{X_2|X_1}(\cdot|\cdot)$, we need $p_{\bar{X}_1|\bar{X}_2}(\cdot|\cdot)$ and $p_{\bar{X}_2|\bar{X}_1}(\cdot|\cdot)$ in Algorithm 1 to evaluate the corresponding confidence interval.

THESIS 1.5: *I derive analytically the false negative error probability of the Attack Detection Algorithm assuming an attacker that can modify sample elements by adding an offset to them, where the offset values are independent and identically distributed according to the normal distribution with arbitrary parameters. Moreover, using the false negative error probability, I derive analytically the distortion caused by the attacker at the output of the Enhanced Data Aggregation Algorithm. [J1] [C1] [P2]*

The false negative error probability β can be defined based on the particular probabilities as

$$\beta = \sum_{j=0}^t P(t_1 = j) \beta^{(j, t-j)} \quad (8)$$

where

$$P(t_1 = j) = \frac{\binom{t}{j} \binom{\frac{n}{2}-j}{\frac{n}{2}}}{\binom{\frac{n}{2}}{\frac{n}{2}}} \quad (9)$$

is the hypergeometric distribution with parameters n , t , and $\frac{n}{2}$. The $\beta^{(j, t-j)}$ particular error probabilities are defined as

$$\beta^{(t_1, t_2)} = \frac{1}{2} (\beta^{(1)} + \beta^{(2)}) \quad (10)$$

where the (t_1, t_2) superscript means that the first half of the sample contains t_1 compromised elements, while the second half contains t_2 compromised elements ($t = t_1 + t_2$). $\beta^{(t_1, t_2)}$ is the average of two particular error probabilities corresponding to the cases of the different condition choice (see Algorithm 1). These particular error probabilities can be defined as

$$\beta^{(1)} = \int_{-\infty}^{\infty} \int_{b_1(\bar{x}_{h,1})}^{b_2(\bar{x}_{h,1})} p_{\bar{X}_{h,2}, \bar{X}_{h,1}}(u, v) du dv \quad (11)$$

$$\beta^{(2)} = \int_{-\infty}^{\infty} \int_{b_1(\bar{x}_{h,2})}^{b_2(\bar{x}_{h,2})} p_{\bar{X}_{h,1}, \bar{X}_{h,2}}(u, v) du dv \quad (12)$$

where subscript h denotes that some elements may be compromised.

The distortion at the output of the Enhanced Data Aggregation Algorithm can be expressed as

$$\begin{aligned}
d(Y|A = 1) &= E[|Y - \hat{Y}|^2|A = 1] = \\
&= E[|Y - \hat{Y}|^2|A = 1, D = 1] \cdot (1 - \beta) + E[|Y - \hat{Y}|^2|A = 1, D = 0] \cdot \beta \\
&= E|Y_{extr} - \hat{Y}|^2 \cdot (1 - \beta) + \frac{1}{n^2}(\tilde{\mu}^2 + \tilde{\sigma}^2) \cdot \beta
\end{aligned} \tag{13}$$

where $A = 1$ means that there is an attack. Assuming that $E|Y_{extr} - \hat{Y}|^2$ is close to zero, I can characterize the distortion as

$$d(Y|A = 1) \cong \frac{1}{n^2}(\tilde{\mu}^2 + \tilde{\sigma}^2) \cdot \beta \tag{14}$$

THESIS 1.6: *I compare the distortion achieved by the Attack Detection Algorithm proposed in Thesis 1.4 to the Maximum Likelihood Decision in the case when the sample consists of two sensor readings and the distribution of the readings is the standard normal distribution. My results show that the difference between the distortions achieved by the two algorithms decreases as the correlation of the sensor readings increases. When the correlation is close to 1, the two algorithms achieve almost the same distortion. [J1]*

The Maximum Likelihood Decision is not applicable in my data and attacker model without further assumptions, however, its importance in decision theory lead me to compare its efficiency to the efficiency of Algorithm 1 in a significantly restricted model. The restriction is the following: I assume that the attacker's distribution is *a priori* known. I emphasize that this assumption is required for the Maximum Likelihood Decision to be able to operate, and it should not be confused with the assumption about the normality made only in order to perform the analysis of my approach; the Attack Detection Algorithm does not need to know the attacker's distribution while the Maximum Likelihood Decision needs. For sake of simplicity, I assume that the attacker's distribution is the Gaussian distribution with known expected value and variance.

To be able to observe the effect of the Maximum Likelihood Decision on the distortion, I have put it in the Enhanced Data Aggregation Algorithm in place of $Det(\cdot, \cdot)$. Figure 4 shows the results of the comparison of the Attack Detection Algorithm and the Maximum Likelihood Decision under a Gaussian data model, both as a building block in the Enhanced Data Aggregation Algorithm. The corresponding values for the calculations are $\mu = 0$, $\sigma = 1$, $\tilde{\sigma} = 1$, and d_{imp} is defined as:

$$\begin{aligned}
d_{imp} &= d(Y|A = 1, D = 0) - d(Y|A = 1) \\
&= \frac{1}{4}(\tilde{\mu}^2 + \tilde{\sigma}^2) - \left[E|Y_{extr} - \hat{Y}|^2 \cdot (1 - \beta) + \frac{1}{4}(\tilde{\mu}^2 + \tilde{\sigma}^2)\beta \right] \\
&\cong \frac{1}{4}(\tilde{\mu}^2 + \tilde{\sigma}^2) \cdot (1 - \beta)
\end{aligned} \tag{15}$$

As one can see in Figure 4, the improvement in the distortion implied by the Maximum Likelihood Decision is higher than for the Attack Detection Algorithm in case of low correlation, however, the difference becomes very small if the correlation is higher. This difference is based on the fact that the Maximum Likelihood Decision takes advantage of the knowledge of the distribution of the attacker's offset. Therefore, in this comparison, where this distribution is assumed to be known to the Maximum Likelihood Decision algorithm, this latter can perform better than the Attack Detection Algorithm. However, if the correlation is higher, the Attack Detection Algorithm performs as well as Maximum Likelihood Decision, even without relying on this extended knowledge.

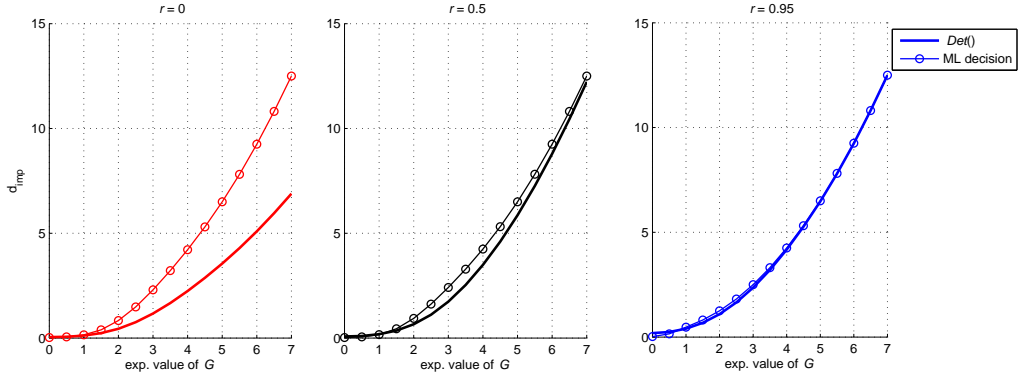


Figure 4: Comparison of Maximum Likelihood Decision and the Attack Detection Algorithm

THEESIS 1.7: *The Attack Detection Algorithm relies on the knowledge of the conditional p.d.f's $p_{X_1|X_2}$ and $p_{X_2|X_1}$ where X_1 and X_2 are representing the sensor readings of a two-element sample. I model and analyze the imprecise knowledge of these conditional p.d.f's and I show that it does not result in a significantly different distortion compared to the case when the knowledge is perfect assuming that the imprecision is moderate. [J1]*

Let us assume that the Attack Detection Algorithm knows only $\hat{p}_{X_1|X_2}(x|y) = p_{X_1|X_2}(x|y) + \Delta(x|y)$ and similarly for $\hat{p}_{X_2|X_1}(\cdot|\cdot)$, where $\int_{-\infty}^{\infty} |\Delta(x|y)|dx < \delta$ for any given y . Moreover, since $p_{X_1|X_2}(\cdot|\cdot)$ and $\hat{p}_{X_1|X_2}(\cdot|\cdot)$ are both probability density functions, $\int_{-\infty}^{\infty} \Delta(x|y)dx = 0$ for any y . The imprecise knowledge implies a wider confidence interval in Algorithm 1 with upper and lower bounds $\hat{b}_1(\cdot)$ and $\hat{b}_2(\cdot)$.

As $\int_{-\infty}^{\infty} \Delta(x|y)dx = 0$, Δ has positive and negative domains as well. Moreover, the integral of the positive domains is equal to the integral of the absolute value of the negative domains. The worst case happens (i.e., $|\hat{b}_i(\cdot) - b_i(\cdot)|$ is the largest) when the positive domains are smaller than $\hat{b}_1(\cdot)$ or greater than $\hat{b}_2(\cdot)$, while all the negative domains are between $\hat{b}_1(\cdot)$ and $\hat{b}_2(\cdot)$. Equally weakening both sides of the confidence interval means putting the same "weight" below $\hat{b}_1(\cdot)$ and above $\hat{b}_2(\cdot)$. Instead of equations

$$\int_{-\infty}^{b_1(z)} p_{X_2|X_1}(u|z)du = \frac{\alpha}{2} \quad (16)$$

$$\int_{b_2(z)}^{\infty} p_{X_2|X_1}(u|z)du = \frac{\alpha}{2} \quad (17)$$

$$\int_{-\infty}^{b_1(x_2)} p_{X_1|X_2}(u|x_2)du = \frac{\alpha}{2} \quad (18)$$

$$\int_{b_2(x_2)}^{\infty} p_{X_1|X_2}(u|x_2)du = \frac{\alpha}{2} \quad (19)$$

this would imply

$$\int_{-\infty}^{\hat{b}_1(z)} p_{X_2|X_1}(u|z)du = \frac{\alpha}{2} - \frac{\delta}{4} \quad (20)$$

$$\int_{\hat{b}_2(z)}^{\infty} p_{X_2|X_1}(u|z)du = \frac{\alpha}{2} - \frac{\delta}{4} \quad (21)$$

$$\int_{-\infty}^{\hat{b}_1(x_2)} p_{X_1|X_2}(u|x_2)du = \frac{\alpha}{2} - \frac{\delta}{4} \quad (22)$$

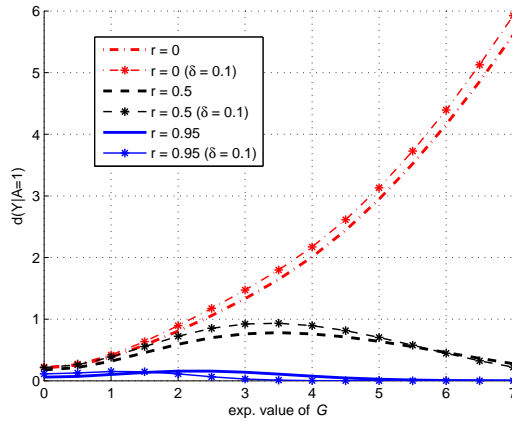


Figure 5: The effects of the imprecise knowledge of the p.d.f's on the distortion

$$\int_{\hat{b}_2(x_2)}^{\infty} p_{X_1|X_2}(u|x_2)du = \frac{\alpha}{2} - \frac{\delta}{4} \quad (23)$$

where α is the false positive probability. Using these formulas one can calculate the new confidence interval bounds $\hat{b}_1(\cdot)$ and $\hat{b}_2(\cdot)$, and with those one is able to evaluate the effect of the imprecise knowledge of the conditional p.d.f's on the distortion. (I note, however, that Equations (20)-(23) implicitly upper bound δ by 2α .) Figure 5 shows the results of this evaluation for $\delta = 0.1$ and $n = 2$. As expected, the imprecise knowledge of the conditional p.d.f's usually implies weaker attack detection capabilities, however, these calculations belong to the worst case (i.e., for a specially constructed Δ). The interesting news of the figure is that shifting the bounds of the confidence interval does not necessarily results in a higher distortion for correlated measurements.

THESIS 1.8: *The Attack Detection Algorithm relies on the fact that the correlation coefficient among the sample elements is constant. I analyze formally the case of distance-dependent correlation assuming the Power Exponential correlation model [BOS01] and I show that it does not result in a significantly different distortion compared to the case when the correlation coefficient is constant and its value is 0.95. [J1]*

Until now, I have assumed that the correlation coefficient r has the same value for all pairs of readings. In reality, every pair of readings has a specific correlation value which depends on the distance of the nodes that produced the readings, and on some physical properties of the environment in which the nodes are deployed. The most widely used correlation model in the literature on spatial statistics is the Power Exponential model [GGG07, WT93] with several applications [Stu01, VA06, VAA04, AVA04, Rap01, BKK06], therefore I applied it as well.

The results of the analysis of the effect of non-constant correlation are very interesting. The resulting values for the *improvement in the distortion* are nearly the same as in the case of constant correlation when $r = 0.95$. The improvement in the distortion is defined in Equation (15) (see Thesis 1.6). Table 1 shows a comparison of numerical values for $t = 2$.

This small difference between the d_{imp} values of the two cases clearly shows, on the one hand, that one is able to model the pairwise correlation among the sample elements with a fixed correlation coefficient in the long run. This, on the other hand, reinforces my previous results: even though I used a simplified scheme in which I considered the correlation coefficient to be constant (with two describing values of 0.95 and 0.5, and the value of 0 for the independent case), my results are still highly relevant

Table 1: Numerical values of the $r = 0.95$ curve compared to the d_{imp} values in case the correlation coefficient is not constant

d_{imp} for $r = 0.95$	d_{imp} for r_{ij}
0.0046	0.0048
0.0115	0.0122
0.0342	0.0345
0.0700	0.0716
0.1192	0.1199
0.1823	0.1852
0.2595	0.2741
0.3508	0.3587

when we consider the more realistic scenario of distance-dependent correlation coefficients among the sample elements.

THESIS 1.9: *The attacker is modelled to add offsets to some of the sensor readings, where the offsets are independent and identically distributed random variables. I investigate, by means of simulations, the case when the attacker is more sophisticated: he can arbitrarily modify the observed sample elements. I show that the results for distortion achieved by the sophisticated adversary are highly related to the analytical results in Thesis 1.5. [J1]*

I assume that the attacker knows the Enhanced Data Aggregation Algorithm, including the Attack Detection Algorithm $Det(\cdot, \cdot)$. Moreover, the attacker also knows the size of the sample that the base station gathers in a given query, he can observe some of the sample elements, and he can arbitrarily modify these observed elements. However, the attacker is not assumed to know the exact way of halving that is applied to shrink the sample of size n into two elements.

This sophisticated attacker is able to choose the best attack in the long run after estimating the unobserved (unknown) elements of the sample. This can be done as follows. At first, the attacker analyzes the observed sample part and gives an estimation on the remaining elements (the attacker is able to do this since he knows the size of the gathered sample). This estimation can be of any kind, for the simulations below I used the method to replace every unknown element with the average of the observed elements. Then, the attacker is able to investigate all the possible halvings and calculate the distortion for them for each possible value of the offset parameter, which parameter is under the control of the adversary. I note that the attacker is not restricted to compromise all the observed measurements, but he is able to choose the number of measurements to compromise in the range $[1, t]$, where t is the number of observed elements in this case. The attacker selects those measurements to compromise, the modification of which leads to the highest distortion on average.

As the attacker does not know the sample halving procedure, the highest distortion on average is calculated by averaging the individual distortions over the different halvings (all the halvings have equal probability in $Det(\cdot, \cdot)$, which is $\frac{1}{2^t}$) and taking the maximum of the resulting vector.

In the first three subfigures in Figure 6 (i.e., up to 30% compromised nodes, as $n = 10$), the highly correlated measurements imply smaller distortion than the independent measurements. The last two subfigures, however, show that the effect of a powerful attacker, who can compromise the measurement of a high number of nodes, is better eliminable when the sensor readings are independent. All the same, low correlations (like $r = 0.5$) usually weaken the capabilities of the proposed solution. In a realistic attack scenario (i.e., where the attacker is only able to compromise the measurement of a small number of sensor nodes) the distortion of the Enhanced Data Aggregation Algorithm can grow up to 2.5σ for less correlated and independent samples, while it usually stays below 1.2σ for highly correlated samples and for $\alpha = 0.1$.

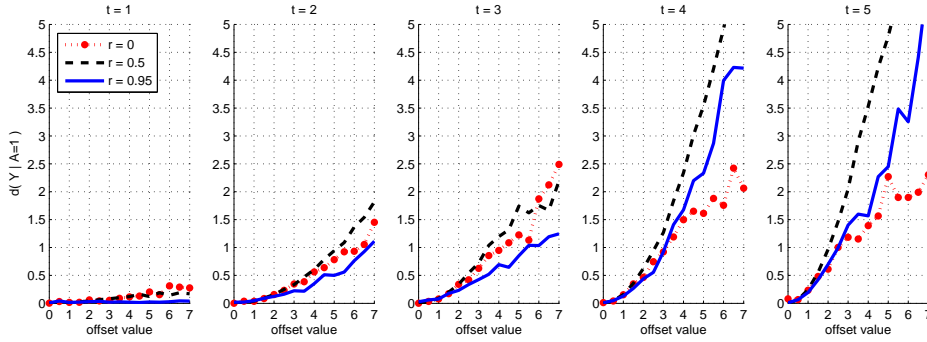


Figure 6: Distortion caused by a sophisticated adversary for different values of the correlation coeff. r

The results for the distortion caused by a sophisticated attacker can be summarized as these are highly related to the analytical results in Thesis 1.5 considering the form and the position of the related curves, however, a sophisticated attacker can achieve a higher distortion than the previously considered simplified attacker.

4.2 RANBAR: RANSAC-based Resilient Aggregation in Sensor Networks

In order to mitigate the environment altering attack, I present a novel resilient data aggregation technique based on sample filtering. This technique is called RANBAR and it is based on the RANSAC (RANdom SAMple Consensus) paradigm [FB81], which is well-known e.g., in computer vision [LPT00]. The RANSAC paradigm gives us a hint on how to instantiate a model if there are a lot of compromised data elements. However, the paradigm does not specify an algorithm and it uses a guess for the number of compromised elements, which is not known in general in real life environments.

THESES 2: *I present and investigate a novel resilient data aggregation technique designed for sensor networks, called RANBAR, that it is based on the RANSAC paradigm. [B1] [C3] [N1]*

RANSAC defines a principle for filtering non-consistent data from a sample, or in other words, fitting a model to experimental data. The principle of RANSAC is the opposite to that of conventional smoothing techniques: rather than using as much of the data as possible to obtain an initial solution and then attempting to eliminate the non-consistent data elements, RANSAC uses as few of the data as feasible to determine a possible model and then tries to enlarge the initial data set with the consistent data.

THESES 2.1: *I present a new resilient data aggregation algorithm, called RANBAR, which follows the RANSAC paradigm and works under the assumption that the sensor readings are independent and identically distributed. Moreover, I determine, by means of empirical analysis, the best trade-off values of the parameters of RANBAR under the assumption that the sensor readings come from a normal distribution with unknown parameters. [B1] [C3] [N1]*

The operation of the RANBAR algorithm is as follows (see Algorithm 3). The base station receives the sample compromised previously by the attacker. The sample is the input of the RANBAR algorithm. First, a set S of minimum size will be randomly chosen to establish a preliminary model. The size of set S is s , and the model M is the probability density function $p(x)$ of the Gaussian distribution with empirical mean $\hat{\theta} = \frac{1}{s} \sum_{i=1}^s S_i$ and with empirical variance $\hat{\sigma}^2 = \frac{1}{s-1} \sum_{i=1}^s (S_i - \hat{\theta})^2$. S_i denotes the i th element of set S .

The consistency check in line 4 is done by Algorithm 4.

The remaining sample elements after the operation of Algorithm 4 constitute the set S^* , called also the consensus set of S . If the size of S^* is smaller than a required size q then the algorithm starts again

Algorithm 3 RANBAR Pseudo-Algorithm

```
1: while No. of trials  $\leq$  Max trials do
2:   Randomly select  $s$  data elements ( $S$ )
3:   Instantiate the model  $M$  based on  $S$ 
4:   Select all data elements within some error tolerance from  $M(S^*)$ 
5:   if  $\#(S^*) > threshold$  then
6:     Instantiate the model  $M^*$  based on  $S^*$ 
7:     return
8:   end if
9: end while
10: Compute  $M^*$  on the largest  $S^*$  or terminate in failure
```

Algorithm 4 RANBAR Consistency Check

```
1: repeat
2:   (re)calculate the histogram of the elements
3:   calculate the distance between the p.d.f. of model  $M$  (denoted by  $p(x)$ ) and the histogram of the
   sample (denoted by  $h(x)$ ), where the distance is defined by
   
$$d = \int |h(x) - p(x)|_+ dx, \text{ where } |x|_+ = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

4:   drop one element from the bin of the histogram corresponding to the maximum  $|h(x) - p(x)|_+$ 
5: until  $d > \epsilon$ 
```

from the first step, otherwise S^* will be forwarded to the aggregator. There is an upper bound on the maximum number of retrials denoted by f . If there were more iterations than f , the algorithm ends with failure.

The RANBAR algorithm has four parameters that have not been defined yet. Two of these are defined by empirical analysis, since the complicity and the probabilistic nature of the algorithm impeded the formal analysis.

The size s of the initial set is desired to be as small as possible according to the RANSAC paradigm. For the RANBAR algorithm, we need to establish the theoretical histogram of a Gaussian distribution. The Gaussian distribution has two parameters, the expected value θ and the standard deviation σ . A rough estimate for the expected value can be based on a single element from the sample. For an estimate on the standard deviation we need at least two elements. This was the motivation for the choice

$$s = 2 \tag{24}$$

The required size q of the consensus set is the most important parameter of the algorithm. However, the RANSAC paradigm does not give us any hint about the correct choice of its value. If q is small, then the algorithm has a higher probability to succeed, but the aggregate at the end will contain a high level of error caused by an attacker. If q is too big, the algorithm cannot work because of the high demand on the number of elements in the final set. In general, we have no information about the percentage of compromised nodes, but we require the algorithm to work even in extreme situations, i.e., when only half of the network is uncompromised. That is why I have chosen

$$q = \frac{n}{2} \tag{25}$$

where n is the total number of sensor nodes in the network.

The value of f is determined with empirical analysis. I have tested the algorithm with a couple of f values and I have found that the choice of f does not really affect the distortion of the final aggregate, but there is a trade-off between f and the probability of finding a good consensus set S^* . If f is small then

there is a great risk that the algorithm will not find a suitable model, and by increasing f this probability decreases, however, the running time increases. Based on empirical analysis, the choice of

$$f = 15 \quad (26)$$

seems to be convenient.

The error tolerance ϵ is defined as the stopping criterion of the algorithm. When d becomes smaller than ϵ , the repetitive phase of the algorithm ends. Thus, ϵ can be considered as an accuracy level requirement. If ϵ is too big, the output of the algorithm can be far from the real expected value of the uncompromised sample. If ϵ is too small, the algorithm may not shape $h(x)$ to be enough close to $p(x)$, thus it ends with failure. A suitable error tolerance level ϵ can be obtained by testing this metric in the unattacked case and for different proportions of compromised nodes. In the test cases, I have tested all the reasonable ϵ values for some typical attack strengths (denoted by κ), and I have preferred those ϵ values by which both the average and the variance of the distortion was small. A value compatible with all choices of κ is

$$\epsilon = 0.3 \quad (27)$$

THESIS 2.2: *I show, by means of simulation, that the breakdown point of the algorithm in Thesis 2.1 is 0.5. Moreover, I also show that when the proportion of compromised sensor readings is close to the breakdown point, the algorithm in Thesis 2.1 achieves a smaller distortion than the median. All simulations related to this thesis are performed under the assumption that the sensor readings are independent and identically distributed, they come from a normal distribution, and the attacker's strategy is to alter the compromised readings to an arbitrary common value. [B1] [C3] [N1]*

In Figure 7, I have compared the resilient aggregation methods suggested by Wagner in [Wag04] (i.e., trimming and median) with RANBAR in case of the mentioned attacker. The horizontal axis corresponds to different attack strengths, the vertical axis corresponds to the distortion of the algorithms. The stipple line shows how the 5% trimming method performs. The 5% trimming has a breakdown point of 0.05. Of course, the trimming level can be lifted up to 50%, but with this the accuracy of the method declines. Thus, there is a need to precisely foretell the proportion of compromised readings the method has to encounter, but in a real life situation this information is usually not available.

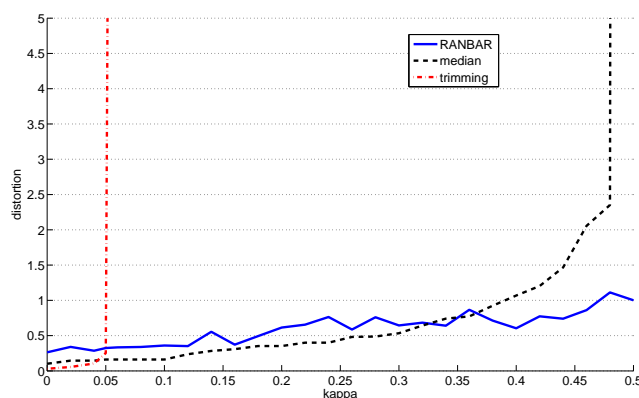


Figure 7: Comparison of RANBAR, median and the trimming calculation

The results of the median calculations for different values of κ are represented with the dotted line. The lesson of Figure 7 is that the median calculation and the RANBAR algorithm produce similarly distorted estimates for $\kappa < \frac{1}{3}$, but for higher κ values the results of the median calculation rapidly

decline, while the results of RANBAR are still close to the real average of the original sample. The explanation for this is that using the median method, one compromised sample element can alter the result by changing it to its neighbor in the row of sorted elements. For more compromised elements the result can be as many indices far from the real median as many compromised readings are presented in the sample. Supposing a normally distributed sample by which the majority of elements are located closely to the real median, the attacker has to compromise about one third of the sample to reach a small distortion, but after this, each compromised node can spoil the median significantly. In contrast to this, the result of RANBAR does not diverge notably from the real average even for high values of κ . Recall that for example $\kappa = \frac{1}{3}$ does not mean that the attacker controls everything in $\frac{1}{3}$ of the network (e.g., he may cannot disrupt the communication protocols), but he is able to alter the readings of $\frac{1}{3}$ of the sensors. (I note that Figure 7 does not show a breakdown of RANBAR, as the corresponding curve simply ends at $\kappa = 0.5$. The reason for this is that the distortion of RANBAR remains bounded until the latter point, but above this value the algorithm fails because of the high value of the required size of the consensus set.)

As one can see, the distortion of RANBAR is always upper bounded for $\kappa < 0.5$, in other words, the breakdown point of the RANBAR algorithm is 0.5 considering this specific attacker. In addition, the distortion usually stays below σ , which means that the attacker has very limited influence even if he is able to compromise about half of the network.

THESIS 2.3: *I analytically derive the optimal attack against the algorithm in Thesis 2.1, where optimality means that the attacker is able to arbitrarily distort the output of the algorithm in Thesis 2.1 with the minimum number of compromised sample elements. Moreover, I analytically derive the breakdown point of the algorithm in Thesis 2.1 considering the optimal attack. The derivations of this thesis are performed under the assumption that the sensor readings are independent and identically distributed, and they come from a normal distribution.*

First, I define some important notions:

Definition 1. Original distribution is the distribution of the unattacked sample.

Definition 2. Target distribution is the distribution that the attacker wants for RANBAR to accept as model M^* .

Definition 3. Supporting element is an element (whether attacked or unattacked) that, when chosen into set S^* , fits a given distribution.

The optimal attacker attacks by inventing an appropriate target distribution, and setting the compromised elements to values the choice of which into S would result in the parameters of the invented target distribution during the instantiation of model M . Of course, the attacker cannot influence the choice of S , but there is a definite probability of the event that appropriate elements of the target distribution are chosen.

As the optimal attacker wants to attack the minimum number of elements, it has to use some elements of the original sample as supporting elements of its target distribution in order to satisfy the criterion regarding the minimum sample size after the consistency check, which is $\frac{n}{2}$, where n is the size of the original sample. Therefore, the attacker has to construct its target distribution so that it overlaps with the original distribution, otherwise the original elements could not be fitted under the target distribution.

Lemma 1. *If the attacker constructs its target distribution in a way that at least three of its bins overlap with the original distribution (considering the truncation at the confidence interval), then the distortion achievable by the attacker is upper bounded.*

Lemma 2. *If the attacker constructs its target distribution in a way that at most one of its bins overlaps with the original distribution (considering the truncation at the confidence interval), then $\#(S^*) \geq q$ never holds if $\kappa < 0.25$.*

Corollary 1. *If the attacker wants to perform its optimal attack by compromising fewer than $\kappa = 0.25$ proportion of the original sample, then it has to construct its target distribution in a way that exactly two of its bins overlap with the original distribution.*

Lemma 3. *The best choice of the attacker for the two bins overlapping with the original distribution is the two largest bins, i.e., the two bins closest to the expected value of the target distribution.*

Definition 4. The a priori area x_{prior} in a bin is the frequency of elements in that bin before the operation of Algorithm 4, while the a posteriori area x_{post} in a bin is the frequency of elements in that bin after the operation of Algorithm 4.

One can calculate the maximum a priori area in the largest bins considering the normalization step in Algorithm 4 based on the equation

$$x_{post} = \frac{nx_{prior} - cut_num}{rem} \quad (28)$$

where cut_num is the number of elements dropped from that bin, and rem is the total number of remaining elements. The result of this calculation is that $x_{prior} \leq 0.42$ for both of the largest bins. As the attacker has to be able to compromise the elements that would not fit into the latter bins, the minimum proportion of elements it has to compromise is $1 - 2 \cdot 0.42 = 0.16$. However, with $\kappa = 0.16$ the attacker can already achieve an arbitrarily high distortion.

Theorem 1. *The optimal attack against RANBAR requires $\kappa = 0.16$.*

As one can see from Theorem 1, the breakdown point of RANBAR in case of an optimal attack is 0.16, and it depends on the area in the bins and on the error tolerance ϵ .

Theorem 2. *The breakdown point of RANBAR considering the optimal attack is at least $\frac{1}{2} - \frac{\epsilon}{2} - V$, where V is the area of the largest bin.*

Without substantial modification of the algorithm the breakdown point can be increased up to 0.5, which is the theoretical maximum. The breakdown point can be lifted if one decreases the area of the bins (i.e., applies more than 10 bins) or/and decreases the value of ϵ . When decreasing only the area of the bins, the breakdown point tends to 0.35, while with the decrease of ϵ simultaneously, the breakdown point tends to 0.5 as

$$\lim_{\substack{V \rightarrow 0 \\ \epsilon \rightarrow 0}} \left(\frac{1}{2} - \frac{\epsilon}{2} - V \right) = 0.5 \quad (29)$$

As a consequence, RANBAR can achieve a breakdown point of 0.5, even when assuming the optimal attack against it.

4.3 PANEL: Position-based Aggregator Node Election in Sensor Networks

Beside resilient data aggregation in sensor networks, I also deal with the problem of aggregator node election in sensor networks. As sensor nodes are often severely resource constrained, various techniques have been proposed to ensure the efficient operation of sensor networks. One of these techniques is called *aggregation* or *in-network processing*. The idea is that instead of forwarding (in case of synchronous applications) or storing (in case of asynchronous applications) raw sensor readings, data can be first processed, combined, and compressed by some distinguished sensor nodes, called *aggregators*.

While aggregation increases the overall efficiency of the sensor network, the aggregator nodes themselves use more resources than the regular sensor nodes. For this reason, it is desirable to change the aggregators from time to time, and thereby, to better balance the load on the sensor nodes. For this purpose, aggregator node election protocols can be used in the sensor network that allow dynamic re-assignment of the aggregator role.

THESES 3: *I present and analyze a new energy efficient position-based aggregator node election protocol developed for wireless sensor networks, called PANEL. [J2] [C2] [P1]*

The novelty of PANEL with respect to other aggregator node election protocols is that it supports asynchronous sensor network applications where the sensor readings are fetched by the base stations after some delay. In particular, the motivation for the design of PANEL was to support reliable and persistent data storage applications, such as TinyPEDS [GWMA06]. PANEL ensures load balancing, and it supports intra- and inter-cluster routing allowing sensor to aggregator, aggregator to aggregator, base station to aggregator, and aggregator to base station communications.

THESIS 3.1: *I present a new position-based aggregator node election protocol, called PANEL. PANEL ensures load balancing in the sense that each node is elected as aggregator nearly equally frequently. Moreover, PANEL also establishes multi-hop data forwarding routes for the cluster member nodes towards the aggregator node. [J2] [C2] [P1]*

One of the main assumptions that PANEL relies on is that the sensor nodes are static and they are aware of their geographical position. The base stations may not necessarily be static, but they can be mobile and their presence can be sporadic. The sensor nodes are deployed in a bounded area, and this area is partitioned into geographical clusters. The clustering is determined before the deployment of the network, and each sensor node is pre-loaded with the geographical information of the cluster which it belongs to. For simplicity, I assume that the deployment area is a rectangle, and the clusters are equal sized squares. I aim at electing a single aggregator per cluster. I assume that the density of the network is large enough so that the nodes within each cluster are connected when they use maximum power for transmission. Finally, I assume that time is divided into epochs, and the nodes are synchronized such that each of them knows when a new epoch begins.

At the beginning of each epoch, a reference point \vec{R}_j is computed in each cluster j by every node in a completely distributed manner. In fact, the computation of the reference point depends only on the epoch number, and it can be executed by every node independently and locally. Once the reference point is computed, the nodes in the cluster elect the node that is the *closest to the reference point* as the aggregator for the given epoch.

The aggregator node election procedure needs communications within the cluster. PANEL takes advantage of these communications and uses them to establish routing tables for intra-cluster routing. In particular, at the end of the aggregator node election procedure, the nodes also learn the next hop towards the aggregator elected for the current epoch.

PANEL also includes a position-based routing protocol that is used in inter-cluster communications. The position-based routing protocol is used for routing messages from a distant base station or from a distant aggregator towards the reference point of a given cluster. Once the message enters the cluster, it is routed further towards the aggregator using the intra-cluster routing protocol based on the routing tables established during the aggregator node election procedure.

PANEL can also support reliable persistent data storage applications such as TinyPEDS [GWMA06]. Reliability can be achieved by replicating the data aggregated by the aggregator nodes at other aggregator nodes (called backup cluster heads). The routing protocols of PANEL can support this by routing the messages containing the replicated data using PANEL's position-based inter-cluster routing protocol towards the reference point of the selected backup cluster, and then switching to the intra-cluster routing protocol of PANEL to deliver the data to the aggregator of that cluster.

The computation of the reference point consists in calling a pseudo-random function H that maps e to a relative position \vec{Q} inside the cluster. Formally, $H(e) = \vec{Q}$, where $\vec{Q} \in (-\Delta d, d + \Delta d) \times (-\Delta d, d + \Delta d)$, d is the size of the cluster, and $\Delta < 0.5$ is a parameter which I will explain in Thesis 3.2. Once the reference points are computed, the nodes start the aggregator node election procedure according to Algorithm 5.

Algorithm 5 The pseudo-code of the aggregator node election in PANEL

Input:

identifier id_{self} and position \vec{P}_{self} of the node executing the algorithm
parameters \vec{O}_{self} and d of the cluster of the node executing the algorithm
current reference point \vec{R}_{self} of the cluster and epoch number e_{now}
running time T_{elec} of the algorithm

Output:

identifier id_{aggr} and position \vec{P}_{aggr} of the elected aggregator node

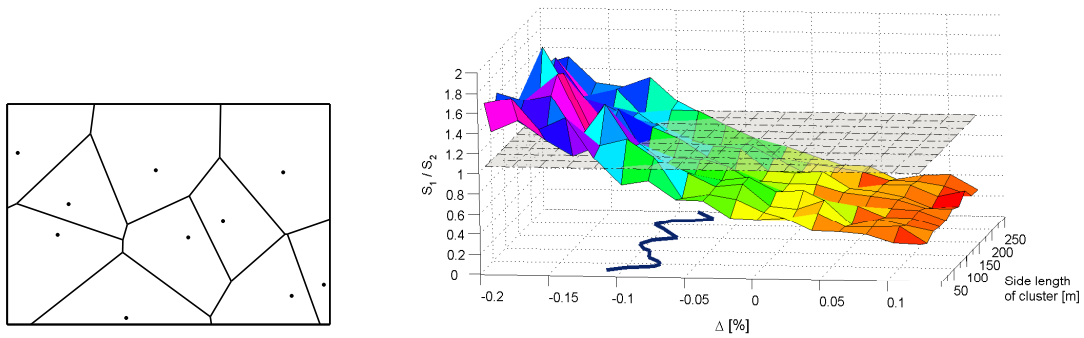
```
set  $id_{aggr} = id_{self}$ ;  
set  $\vec{P}_{aggr} = \vec{P}_{self}$ ;  
set timer  $t_0 = T_{elec}$ ;  
set timer  $t_1 = f(D(\vec{P}_{self}, \vec{R}_{self}))$ ;  
while timer  $t_0$  is still active do  
  wait until timer  $t_1$  fires or an announcement  $m$  is received;  
  case timer  $t_1$  fired:  
    broadcast [announcement |  $e_{now}$  |  $id_{self}$  |  $\vec{P}_{self}$ ] with max power;  
  case an announcement  $m = [\text{announcement} | e | id | \vec{P}]$  is received:  
    if the pair  $(e, id)$  has been seen before then drop  $m$ ;  
    else if  $e \neq e_{now}$  or  $\vec{P} \notin \text{square}(\vec{O}_{self}, d)$  then drop  $m$ ;  
    else if  $D(\vec{P}, \vec{R}_{self}) > D(\vec{P}_{aggr}, \vec{R}_{self})$  then drop  $m$ ;  
    else  
      set  $id_{aggr} = id$ ;  
      set  $\vec{P}_{aggr} = \vec{P}$ ;  
      if timer  $t_1$  is still active then cancel timer  $t_1$ ;  
      re-broadcast  $m$  with max power;  
    end if  
  end while  
output  $id_{aggr}, \vec{P}_{aggr}$ 
```

After some predefined time T_{elec} , the aggregator node election phase is closed, and each node considers the recorded candidate aggregator as the aggregator for the current epoch.

THESIS 3.2: *During the operation of the protocol in Thesis 3.1, the nodes are elected aggregator only nearly equally frequently. I propose an empirical solution to reduce this irregularity in aggregator node election frequency. [J2] [C2]*

I explain now why parameter Δ is needed in the reference point computation in Thesis 3.1, and how its value can be determined. The probability that a given node becomes aggregator in PANEL is determined by the size of the Voronoi cell of the node, and the size of the area within which the reference point is selected. For load balancing purposes, I would like that each node becomes aggregator with nearly the same probability, thus, I would like that the Voronoi cells of the nodes have approximately the same size.

Let us consider Figure 8(a) for illustration of the Voronoi cells of the nodes in a cluster. We can observe a "border effect" in this figure, namely, the size of the Voronoi cells of the nodes close to the edge of the cluster is larger than that of the nodes in the middle. The reason of this phenomenon in one dimension can be most easily explained as follows. In one dimension, the Voronoi cells are intervals on the number line, the size of which is a function of the internode distances, and the size of the first and last interval is also determined by the borders. However, the size of the Voronoi cells of the nodes is



(a) The Voronoi cells of the nodes in a cluster

(b) Determining the value of parameter Δ by simulations

Figure 8: Voronoi cell illustration and computation of the proper value of Δ

determined *not* by the uniformity of the deployment, but by the order statistics of the uniform distribution. One can effectively mitigate this border effect by adjusting the size of the area within which the reference point is selected, as with this, one can adjust the size of the Voronoi cells of the nodes on the edge of the cluster. Parameter Δ expresses the magnitude of this adjusting operation in percent of the original cluster size d . For example, $\Delta = -0.1$ means that on each side of the cluster the bounds are contracted by 10%.

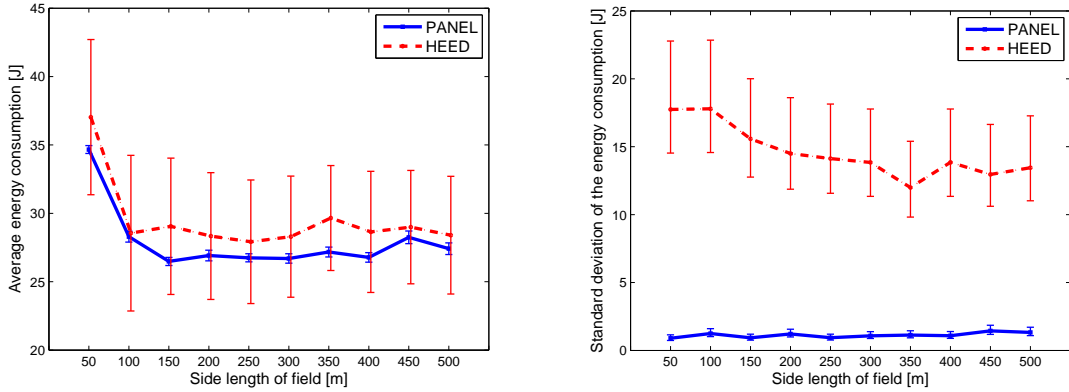
It is not easy to determine an appropriate value for Δ analytically due to the complexity of the computation of the size of the Voronoi cells. Therefore, I propose to determine its value by simulations. In Figure 8(b), on the z axis, I have the ratio between the average size S_1 of the bounded Voronoi cells (i.e., the cells close to the center of the cluster) and the average size S_2 of the unbounded Voronoi cells (i.e., the cells on the edge of the cluster) as a function of parameter Δ and the cluster size given by the side length. The plane at $z = 1$ corresponds to the optimum, where the average sizes of the cells of the two types are equal. The intersection of this plane and the surface obtained by simulations is projected to the $z = 0$ plane. This projected curve gives the optimal value of parameter Δ for different cluster sizes assuming that there are 10 nodes in the cluster. As one can see, the optimal value is usually between -0.12 and -0.07.

THESIS 3.3: *I investigate, by means of simulations, the energy efficiency of the protocol in Thesis 3.1 in comparison with HEED [YF04], a well-known aggregator node election solution. I show that the protocol in Thesis 3.1 is more energy efficient than HEED independently of the node density. [J2]*

I consider 40 uniformly randomly placed nodes in a square shaped field consisting in 4 equal shaped clusters, where the node density is controlled through the size of the field. My simulation scenarios comprise not just cluster head (i.e., aggregator node) election, but data message sending as well. Instead of measuring just the energy spent for cluster head election, in Figure 9(a), I show the total average energy consumed by PANEL and HEED. The horizontal axis corresponds to the field size, while the vertical axis corresponds to the average energy consumption. The solid curve corresponds to PANEL, while the dashed curve corresponds to HEED. The whiskers show the 95% confidence interval of the corresponding values.

As one can see, PANEL consumes less energy in total than HEED, independently from the node density. Moreover, the whiskers in Figure 9(a) show that the energy consumption of PANEL can be forecasted more precisely than the energy consumption of HEED, as the 95% confidence intervals of the energy consumption of PANEL are more narrow than that of HEED. This property is confirmed by Figure 9(b), which shows that the standard deviation of the average energy consumption is much higher

in case of HEED than in case of PANEL. (The whiskers in Figure 9(b) correspond to the 95% confidence interval of the standard deviation of the energy consumption.) I emphasize that the number of rounds (i.e., the amount of data message sending) highly influences my results: the rounds are where PANEL is more energy efficient than HEED, thus, increasing the number of rounds beyond the actual 5 would result in even better performance of PANEL with respect to HEED.



(a) Total average energy consumption as a function of the field size

(b) Standard deviation of the total average energy consumption as a function of the field size

Figure 9: Total energy consumption comparison of PANEL and HEED

THESIS 3.4: *I propose security extensions to the protocol in Thesis 3.1. These extensions help to prevent or mitigate an attacker’s activity aiming at distorting the aggregate at the aggregator node or interfering with the aggregator node election process. [J2] [C2]*

There are several ways how an adversary can spoil the operation of PANEL. In the following, I detail these attacks and propose countermeasures against them. I assume that the attacker is able to capture and reverse-engineer one or more nodes, thus, the attacker is able to control these nodes and has knowledge about all the information stored in these nodes (e.g., secret keys, measurement data, etc.). I note that the formal security analysis of my proposals below are beyond the scope of my dissertation, as it would require an appropriate formal model the development of which itself could be subject of independent research. However, following the common approach, the most common attacks are investigated below.

Distorting the aggregate by environment altering or capturing: The most straightforward type of attack aims at distorting the aggregate at the cluster head. To achieve this, an attacker can either (i) modify the environment of the attacked sensor node, or (ii) capture the sensor node and alter the measured values as desired.

Countermeasures: Both of these attacks can be circumvented using a statistical sample filtering approach at the cluster head (like e.g., RANBAR, see Theses 2). (I note that cryptography cannot help here as these attacks cannot be detected with cryptographic tools.)

Distorting the aggregate by message altering: In order to achieve his goal (i.e., to distort the aggregate), the adversary can also (iii) force the captured node to alter the data field of a forwarded message that comes from another node, or (iv) send false measurement data in the name of other nodes to the cluster head.

Countermeasures: The appropriate tool against these attacks is cryptographic integrity protection (in case of (iii)), and authentication (in case of (iv)). For these, the nodes need for example a public-private key pair and they have to sign their messages using the private key, moreover, they have to attach their

public key to the message after it was signed. (I note that assuming public-key cryptography in sensor networks is not far-fetched according to [PLP06].)

Interfering with the cluster head election process: The attacker can also interfere with the cluster head election process by sending an advertisement message in the name of a node that would not become aggregator based on the position of the reference point.

Countermeasures: The nodes that hear this fake advertisement message can check the validity of the signature on the advertisement message, and can drop messages with invalid signatures.

Manipulating the cluster head election process to become cluster head: Another typical attack against aggregator node election protocols is to manipulate the execution in such a way that the nodes controlled by the adversary become aggregators more frequently than they should (see e.g., [SWAG07]). In this way, the adversary can collect information from the network easier, as nodes send their sensor readings to the aggregators. In PANEL, such an attack can be perpetrated using fake information in the announcement message in the aggregator node election phase by a captured node that uses (i) its correct identifier, but fake position information, or (ii) a fake identifier along with fake position information. Moreover, (iii) the adversary can deploy new nodes at desired positions.

Countermeasures: PANEL can be easily extended with security measures to prevent even these misdeeds. First of all, the base station can use public-key cryptography and sign the nodes' identifier with its private key, and load the corresponding public key on the sensors before deployment. Using this signed identifier, a node that receives an announcement message can check whether it contains a valid identifier or not using the public key of the base station, but no one except the base station is able to generate new identifiers. According to this, the advertisement messages should be extended with the signed identifier, and when a node receives an advertisement message

$$[\text{announcement} \mid \text{epoch} \mid id_{fake} \mid id_{sig_{BS}} \mid pos_{fake} \mid \dots \\ [\text{announcement} \mid \text{epoch} \mid id_{fake} \mid id_{sig_{BS}} \mid pos_{fake}]_{sig} \mid cert]$$

it can check whether $K_{BS}^P(id_{sig_{BS}}) = id_{fake}$, where K_{BS}^P is the public key of the base station, and if not, it can discard the message. This thwarts attacks (ii) and (iii). One can thwart attack (i) by allowing the nodes to keep in their routing tables the position information of the other nodes from which they have already heard an announcement. This information can be kept in the routing tables even beyond the duration of an epoch. Therefore, the nodes can detect if a captured or corrupted node tries to report itself at different positions in different epochs. If the above attack is detected, the detector node can flood an alarm message in the cluster including the cheating node's identifier and the proof for the cheating, i.e., the two advertisement messages with the same identifier and different position information, both signed by the cheating node.

THEESIS 3.5: *The protocol in Thesis 3.1 assumes for its correct operation that the nodes in a cluster are connected. I propose extensions to the protocol in Thesis 3.1 on order to circumvent the evolving problems in case the above assumption does not hold. [J2] [C2]*

A crucial assumption of PANEL is that the nodes within a cluster form a connected subnetwork. If this assumption is not satisfied, and the subnetwork within a cluster is partitioned, then some nodes will not hear the announcement of the node closest to the reference point, and they will elect another node as aggregator.

Solution 1: A possible solution is to extend the area in which an announcement is flooded beyond the borders of the corresponding cluster. For instance, the announcement can also be flooded in the neighboring clusters. This would increase the probability that each node in the corresponding cluster receives

the announcement even if the subnetwork within that cluster is partitioned, because those partitions may be connected through the neighboring clusters. The downside of this approach is the increased energy consumption of the nodes.

Solution 2: A more energy efficient approach is the following. During the aggregator node election phase, we do not extend the area in which an announcement is flooded, and we tolerate that multiple cluster heads are elected. After that, I propose to use the following protocol

$\mathbf{AN}_i \rightarrow \mathbf{BN}_k$: [backup | id_i | $cluster_id_i$ | pos_i | $aggregate_i$]
 $\mathbf{AN}_j \rightarrow \mathbf{BN}_k$: [backup | id_j | $cluster_id_i$ | pos_j | $aggregate_j$]
 \mathbf{BN}_k : detects that $\#(\text{backup messages}) > 1$ and calculates $final_aggregate$
 $\mathbf{BN}_k \rightarrow \mathbf{pos}_i$: [correction | id_k | $final_aggregate$]
 $\mathbf{BN}_k \rightarrow \mathbf{pos}_j$: [correction | id_k | $final_aggregate$]
 $\mathbf{BN}_k \rightarrow \mathbf{BS}$: [notification_of_disconnectivity | id_k | $cluster_id_i$] (optional)

where \mathbf{AN} , \mathbf{BN} , and \mathbf{BS} stand for the aggregator node, the backup node, and the base station, respectively, and \mathbf{pos} as the receiver means that the message has to be sent using position-based routing. $final_aggregate$ is the value that the aggregator nodes have to store finally.

Solution 3: This latter solution is efficient in terms of communication overhead, but sometimes it is not applicable. For example, in case of the average, it works fine, as the average of the averages of two subsamples is equal to the average of the sample consisting in the two subsamples (if the two subsamples are of equal size). However, in case of the median, it does not work. Therefore, for such aggregation functions that need the whole sample to produce the correct output, I propose to use the following method.

$\mathbf{AN}_i \rightarrow \mathbf{BN}_k$: [backup | id_i | $cluster_id_i$ | pos_i | $aggregate_i$]
 $\mathbf{AN}_j \rightarrow \mathbf{BN}_k$: [backup | id_j | $cluster_id_i$ | pos_j | $aggregate_j$]
 \mathbf{BN}_k : detects that $\#(\text{backup messages}) > 1$ and chooses final aggregator
 If node i has been chosen as the final aggregator :
 $\mathbf{BN}_k \rightarrow \mathbf{pos}_i$: [notification_of_disconnectivity | id_k | final_aggregator | id_j | pos_j]
 $\mathbf{BN}_k \rightarrow \mathbf{pos}_j$: [notification_of_disconnectivity | id_k | not_final_aggregator | id_i | pos_i]
 $\mathbf{BN}_k \rightarrow \mathbf{BS}$: [notification_of_disconnectivity | id_k | $cluster_id_i$] (optional)
 $\mathbf{AN}_j \rightarrow \mathbf{pos}_i$: [correction | id_j | $measurements_j$]
 \mathbf{AN}_i : receives $measurements_j$ and calculates $final_aggregate$
 $\mathbf{AN}_i \rightarrow \mathbf{pos}_j$: [correction | id_i | $final_aggregate$]

Here, $measurement_j$ means *all* the measurements of node j that it wants to aggregate.

With this technique, even the partitioned subnetworks will have a consistent view of their cluster, independently from the applied aggregator function. Moreover, queries will receive correct answers independently from the queried cluster head.

5 Application of New Results

To prove the viability of the proposed solutions for resilient data aggregation and aggregator node election in low-end sensor networks, I implemented the RANBAR and the PANEL algorithms in TinyOS 2 [Tin07], the most widely used operating system of sensor nodes. Both of these implementations are part of the final demos of the UbiSec&Sens EU FP6 STReP [EU 08a]. One of the specific applications in the project is vineyard monitoring. On the vineyards, the commonly used sensing tools are meteorological stations. Because of their relatively high price, usually only a limited number of stations is used in a field. However, in this case, the measurements do not reflect the real situation in different parts of the plantation given the large variations in the microclimate on a sparse landscape. Wireless sensor networks are an excellent technology which allows improving this situation. In the final demo of UbiSec&Sens, both RANBAR and PANEL are applied in this scenario consisting of 64 sensor nodes in 4 clusters in the Weingut Georg Naegele vineyard [Wei08] located in Neustadt, Germany.

The other specific application in UbiSec&Sens is roadside monitoring. The main idea here is to equip roads with sensors that gather information about the weather, traffic, and road conditions. This information is then processed and used to dynamically define the speed limit, the optimal routes, or detect abnormal situations (e.g., accidents, fog, snow) and so on. This information is then sent to the cars that can be used to warn the drivers about an imminent danger or about the local speed limit. Moreover, in case of an accident, forensics can enquire the sensor network to gain knowledge of the condition of the road at the moment of the accident. Since sensor readings can be highly correlated in such a small geographical area, aggregation and in-network processing may be advantageous to reduce network traffic. Therefore, in the demo, both RANBAR and PANEL are applied in this scenario as well with 20 nodes (and one additional node mounted to the car) in 4 clusters. The outdoor demonstration took place on an airstrip near Heidelberg, Germany.

As it was highlighted by the demonstrations, the implementations of RANBAR and PANEL, although not fully optimized, turned out to be fully applicable both on TelosB [Cro05b] and MicaZ [Cro05a] motes, and even the interworking of these two type of nodes is seamless. One can find the source code of RANBAR and PANEL under the web address in [EU 08a].

Beside the above examples, there is a high number of possible application areas of sensor networks. In the following, I do not aim at covering all these areas, but I give a short overview on the probably most interesting fields where the algorithms that I developed can be applied.

A promising application area of sensor networks is Critical Infrastructure Protection (CIP). CIP aims at assuring the security of vulnerable and interconnected critical infrastructures, like the energy supply system, the banking system, the emergency services, the transportation system, or even the Internet. It would be nice to have an autonomous, decentralized, resilient, and easily deployable diagnostic system for monitoring the operation of the critical infrastructures, and also for detecting and mitigating possible attacks against them. Sensor networks can be applied here, especially the techniques proposed in Theses 1 and Theses 2 (i.e., CORA and RANBAR, respectively) are of particular importance in such applications, as the reliability of diagnosis results is utmost required. Additionally, Theses 3 (i.e., PANEL) can be applied in CIP as well for data collection by reason of its security extensions. A related research project is WSAN4CIP EU FP7 STReP [EU 08b], in which PANEL will serve as a starting point for the development of secure aggregator node election protocols.

Wildlife monitoring is another primary application field of sensor networks. Observing natural spaces with numerous networked sensors can enable long-term data collection at scales and resolutions that are difficult, if not impossible, to obtain otherwise. The ability of nodes to communicate not only allows information and control to be communicated across the network of nodes, but nodes can cooperate in performing more complex tasks, like statistical sampling, data aggregation, and system health and status monitoring. For this, however, one needs to apply both resilient aggregation solutions, and network management protocols. My resilient aggregation solutions can be applied here to satisfy the former need, while PANEL can be applied to satisfy the latter need. Furthermore, as PANEL achieves reliability

by replicating the measurements on distant nodes, it is also applicable in harsh environments, like deserts or rain forests.

Speaking more generally, the concept of sensor networks is encompassed by the ubiquitous or pervasive computing paradigm. Ubiquitous computing means information processing that has been thoroughly integrated into everyday objects and activities. The best example for ubiquitous computing is perhaps the smart home concept. In a smart home, the environmental controls (like light, heating, airing, etc.) can interact with personal biometric monitors woven in clothing so that illumination, heating and air conditions in a room might be modulated according to the owner's actual necessities. Another common scenario consist in refrigerators aware of their contents, able to both plan a variety of menus from the food actually on hand, and warn users of stale or spoiled food. Ubiquitous computing requires information collection solutions in order to efficiently measure the parameters upon which the decisions are based (e.g., biometric parameters, or just the number of milk bottles, etc.). From my solutions, PANEL can be applied here (see Theses 3), as it is able to collect the measurements of the sensors and create reports.

Vehicular ad hoc networks (VANETs) are another motivation application area of CORA and RANBAR. VANET is a form of mobile ad hoc network, which provides communications among nearby vehicles, and between vehicles and nearby fixed equipment, called roadside equipment. The main goal of VANET is providing safety and comfort for passengers. Each vehicle equipped with communication device will be a node in the ad hoc network and can receive and relay messages of other nodes through the wireless network. For example, an application of VANET is that vehicles can send warning messages at road crossings that they approach it, and thus, the drivers of the oncoming vehicles would know in advance that special attention has to be paid in the crossing. This and similar applications will surely decrease the number of traffic accidents. As there are many cars on the road, the amount of information that vehicles receive can be large, therefore, some kind of aggregation is needed in VANETs. However, an attacker can cause serious accidents by inserting wrong status information reports or falsified measurements. My resilient aggregation techniques are perhaps able to solve the problem of such attacks by mitigating the compromised pieces of information. In other words, CORA and RANBAR may be applicable in VANETs as well.

However, I note that the resilient aggregation solutions presented in my dissertation are not applicable in event-driven sensor networks. Therefore, detection-type applications (like e.g., forest fire detection, intrusion detection) are opposed to apply the presented mechanisms, as these applications focus on the presence of extreme values, and the proposed resilient aggregation mechanisms would simply filter them out.

References

- [AVA04] I. F. Akyildiz, M. C. Vuran, and O. B. Akan. On exploiting spatial and temporal correlation in sensors networks. In *Proceedings of the 2nd Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2004.
- [BKK06] G. Bravos, A. G. Kanatas, and A. Kalis. Lifetime evaluation and spatial correlation effects on wireless sensor networks. In *Proceedings of 15th IST Mobile & Wireless Communications Summit*, 2006.
- [BOS01] J. O. Berger, V. De Oliveira, and B. Sansó. Objective Bayesian analysis of spatially correlated data. *Journal of the American Statistical Association*, 96(456):1361–1374, 2001.
- [Cro05a] Crossbow Corporation. MicaZ Datasheet. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAZ_Datasheet.pdf, 2005.
- [Cro05b] Crossbow Corporation. TelosB Datasheet. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf, 2005.
- [EU 08a] EU FP6 STReP. UbiSec&Sens – Ubiquitous Sensing and Security in the European Homeland. <http://www.ist-ubisecsens.org/>, 2008.
- [EU 08b] EU FP7 STReP. WSan4CIP – Wireless Sensor Networks for the Protection of Critical Infrastructures. <http://www.eurescom.de/activities/EU-Projects/wsan4cip.asp>, 2008.
- [FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [GGG07] T. Gneiting, M. Genton, and P. Guttorp. *Geostatistical Space-Time Models, Stationarity, Separability, and Full Symmetry*. Chapman & Hall/CRC, Boca Raton, FL, USA, 2007.
- [GWMA06] J. Girao, D. Westhoff, E. Mykletun, and T. Araki. TinyPEDS: Tiny persistent encrypted data storage in asynchronous wireless sensor networks. *Elsevier Ad Hoc Networks*, June 2006.
- [LPT00] A. J. Lacey, N. Pinitkarn, and N. A. Thacker. An evaluation of the performance of RANSAC algorithms for stereo camera calibration. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2000.
- [Oli05] David J. Olive. *Applied Robust Statistics*. <http://www.math.siu.edu/olive/ol-bookp.htm>, 2005.
- [PLP06] K. Piotrowski, P. Langendoerfer, and S. Peter. How public key cryptography influences wireless sensor node lifetime. In *Proceedings of the 4th ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN)*, 2006.
- [Rap01] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Saddle River, NJ, USA, 2001.

-
- [Stu01] G. L. Stuber. *Principles of mobile communication (2nd ed.)*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.
- [SWAG07] M. Sirivianos, D. Westhoff, F. Armknecht, and J. Girao. Non-manipulable aggregator node election protocols for wireless sensor networks. In *Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2007.
- [Tin07] TinyOS 2. <http://www.tinyos.net/tinyos-2.x/doc/>, 2007.
- [VA06] M. C. Vuran and I. F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 14(2):316–329, 2006.
- [VAA04] M. C. Vuran, O. B. Akan, and I. F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Elsevier Computer Networks*, 45(3):245–259, 2004.
- [Wag04] D. Wagner. Resilient aggregation in sensor networks. In *Proceedings of the 2nd ACM Workshop on Security of Ad hoc and Sensor Networks (SASN)*, 2004.
- [Wei08] Weingut Georg Naegele Vineyard. <http://www.naegele-wein.de>, 2008.
- [WT93] R. O. Weber and P. Talkner. Some remarks on spatial correlation function models. *Monthly Weather Review*, 121(9):2611–2617, 1993.
- [YF04] O. Younis and S. Fahmy. Distributed clustering in ad hoc sensor networks: A hybrid, energy-efficient approach. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, March 2004.

6 Publication of New Results

Book Chapters

- [B1] L. Buttyán, P. Schaffer, and I. Vajda,
Resilient Aggregation: Statistical Approaches
Book chapter in N. P. Mahalik (ed.): *Sensor Network and Configuration*, Springer, 2007.

Journal Papers

- [J1] L. Buttyán, P. Schaffer, and I. Vajda,
CORA: Correlation-based Resilient Aggregation in Sensor Networks,
Accepted to Elsevier Ad Hoc Networks, September 2008.
- [J2] L. Buttyán, P. Schaffer,
PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks,
Submitted to International Journal of Distributed Sensor Networks, September 2008.
- [J3] L. Buttyán, T. Holczer, and P. Schaffer,
Incentives for Cooperation in Multi-hop Wireless Networks,
Híradástechnika, vol. LIX, no. 3, March 2004, pp. 30–34 (in Hungarian).

International Conference/Workshop Papers

- [C1] P. Schaffer, I. Vajda,
CORA: Correlation-based Resilient Aggregation in Sensor Networks,
In Proceedings of the 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2007), Chania, Greece, October 2007.
- [C2] L. Buttyán, P. Schaffer,
PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks,
In Proceedings of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Pisa, Italy, October 2007.
- [C3] L. Buttyán, P. Schaffer, and I. Vajda,
RANBAR: RANSAC-Based Resilient Aggregation in Sensor Networks,
In Proceedings of the 4th ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN), Alexandria, VA, USA, October 2006.
- [C4] L. Buttyán, P. Schaffer, and I. Vajda,
Resilient Aggregation with Attack Detection in Sensor Networks,
In Proceedings of the 2nd IEEE Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS), Pisa, Italy, March 2006.
- [C5] L. Buttyán, T. Holczer, and P. Schaffer,
Spontaneous Cooperation in Multi-domain Sensor Networks,
In Proceedings of the 2nd European Workshop on Security and Privacy in Ad-hoc and Sensor Networks (ESAS), Springer LNCS 3813, Visegrád, Hungary, July 2005.

National Conference/Workshop Papers

- [N1] L. Buttyán, P. Schaffer, and I. Vajda,
RANBAR: RANSAC-Based Resilient Aggregation in Sensor Networks,
In Proceedings of the HSN Workshop, Balatonkenese, Hungary, May 2006.

Posters

- [P1] G. Ács, P. Schaffer, and L. Buttyán,
PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks,
In Proceedings of the HSN Workshop, Balatonkenese, Hungary, May 2008.
- [P2] P. Schaffer, L. Buttyán, and I. Vajda,
CORA: Correlation-based Resilient Aggregation in Sensor Networks,
In Proceedings of the HSN Workshop, Balatonkenese, Hungary, May 2007.

Theses

- [T1] P. Schaffer,
Investigation of the Emergence of Spontaneous Cooperation in Multi-domain Sensor Networks – The Case of the Common Base Station,
M.Sc. Thesis (advisor: Dr. L. Buttyán), BME, Budapest, Hungary, June 2005 (in Hungarian).

Other

- [O1] T. Holczer, P. Schaffer,
The Evolution of Spontaneous Cooperation in Multi-domain Sensor Networks,
TDK (Students Scientific Conference) paper (advisor: Dr. L. Buttyán), 3. prize, Budapest, Hungary, November 2004 (in Hungarian).

Citations

Below, I list the known independent citations to my publications.

- L. Buttyán, P. Schaffer, and I. Vajda, **Resilient Aggregation with Attack Detection in Sensor Networks**, In Proceedings of the 2nd IEEE Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2006), Pisa, Italy, March 2006.

is cited by

- ◊ C. Castelluccia, C. Soriente, **ABBA: A Balls and Bins Approach to Secure Aggregation in WSNs**, In Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops (WiOPT), 2008.
 - ◊ S. Roy, **Secure Data Aggregation in Wireless Sensor Networks**, Ph.D. Dissertation, George Mason University, USA, 2008.
 - ◊ Y. J. Luo, X. Yang, and X. Zhang, **An Effective Resilient Data Aggregation Algorithm in Wireless Sensor Networks**, In Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), 2007.
 - ◊ S. Roy, S. Setia, and S. Jajodia, **Attack-Resilient Hierarchical Data Aggregation in Sensor Networks**, In Proceedings of the Fourth ACM Workshop on Security of Ad hoc and Sensor Networks (SASN), 2006.
- L. Buttyán, P. Schaffer, and I. Vajda, **RANBAR: RANSAC-Based Resilient Aggregation in Sensor Networks**, In Proceedings of the 4th ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN), Alexandria, VA, USA, October 2006.

is cited by

- ◊ E. D. Cristofaro, J.-M. Bohli, and D. Westhoff, **FAIR: Fuzzy-based Aggregation Providing In-network Resilience for Real-time Wireless Sensor Networks**, In Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec), 2009.
- ◊ S. Roy, M. Conti, S. Setia, and S. Jajodia, **Securely Computing an Approximate Median in Wireless Sensor Networks**, In Proceedings of 4th International ICST Conference on Security and Privacy in Communication Networks (SecureComm), 2008.
- ◊ E.-O. Blass, J. Wilke, and M. Zitterbart, **Relaxed Authenticity for Data Aggregation in Wireless Sensor Networks**, In Proceedings of 4th International ICST Conference on Security and Privacy in Communication Networks (SecureComm), 2008.
- ◊ S. Roy, **Secure Data Aggregation in Wireless Sensor Networks**, Ph.D. Dissertation, George Mason University, USA, 2008.
- ◊ S. Setia, S. Roy, and S. Jajodia, **Secure Data Aggregation in Wireless Sensor Networks**, Book chapter in J. Lopez, J. Zhou (eds.): *Wireless Sensor Network Security*, IOS Press, 2008.
- ◊ J.-M. Bohli, A. Hessler, O. Ugus, and D. Westhoff, **A Secure and Resilient WSN Roadside Architecture for Intelligent Transport Systems**, In Proceedings of the First ACM Conference on Wireless Network Security (WiSec), 2008.
- ◊ B. Sun, L. Osborne, Y. Xiao, and S. Guizani, **Intrusion Detection Techniques in Mobile Ad Hoc and Wireless Sensor Networks**, In *IEEE Wireless Communications*, 2007.

-
- ◇ Y. J. Luo, X. Yang, and X. Zhang, **An Effective Resilient Data Aggregation Algorithm in Wireless Sensor Networks**, In Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), 2007.
 - ◇ R. Di Pietro, P. Michiardi, and R. Molva, **Confidentiality and Integrity for Data Aggregation in WSN Using Peer Monitoring**, Eurecom, Research Report, 2007.
 - L. Buttyán, P. Schaffer, **PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks**, In Proceedings of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Pisa, Italy, October 2007.

is cited by

- ◇ E. D. Cristofaro, J.-M. Bohli, and D. Westhoff, **FAIR: Fuzzy-based Aggregation Providing In-network Resilience for Real-time Wireless Sensor Networks**, In Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec), 2009.
- ◇ E. Meshkova, J. Riihijarvi, F. Oldewurtel, C. Jardak, and P. Mahonen, **Service-Oriented Design Methodology for Wireless Sensor Networks: A View through Case Studies**, In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2008.
- ◇ C. Jardak, E. Osipov, and P. Mahonen, **Distributed Information Storage and Collection for WSNs**, In Proceedings of the Fourth IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), 2007.
- L. Buttyán, T. Holczer, and P. Schaffer, **Spontaneous Cooperation in Multi-domain Sensor Networks**, In Proceedings of the 2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS), Springer LNCS 3813, Visegrád, Hungary, July 2005.

is cited by

- ◇ A. G. Forte, H. Schulzrinne, **Cooperation Between Stations in Wireless Networks**, In Proceedings of the IEEE International Conference on Network Protocols (ICNP), 2007.

The total number of known independent citations is 19. The above list does not contain independent citations in project deliverables and dependent citations.