Ad Hoc Networks 7 (2009) 1035–1050

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

CORA: Correlation-based resilient aggregation in sensor networks

Levente Buttyán, Péter Schaffer *, István Vajda

Laboratory of Cryptography and Systems Security (CrySyS), Department of Telecommunications, Budapest University of Technology and Economics, 1117 Budapest, Hungary

ARTICLE INFO

Article history:
Received 20 November 2007
Received in revised form 21 August 2008
Accepted 18 September 2008
Available online 1 October 2008

Keywords:
Sensor networks
Resilient aggregation
Correlation
Attack detection

ABSTRACT

In this paper, we consider the problem of resilient data aggregation in sensor networks, namely, how to aggregate sensor readings collected by the base station when some of those sensor readings may be compromised. Note that an attacker can easily compromise the reading of a sensor by altering the environmental parameters measured by that sensor. We present a statistical framework that is designed to mitigate the effects of the attacker on the output of the aggregation function. The main novelty of our approach compared to most prior work on resilient data aggregation is that we take advantage of the naturally existing correlation between the readings produced by different sensors. In particular, we show how spatial correlation can be represented in the sensor network data model, and how it can be exploited to increase the resilience of data aggregation. The algorithms presented in this paper are flexible enough to be applied without any special assumption on the distribution of the sensor readings or on the strategy of the attacker. The effectiveness of the algorithms is evaluated analytically considering a typical attacker model with various parameters, and by means of simulation considering a sophisticated attacker.

1. Introduction

Wireless sensor networks are considered as a promising technology that has a wide range of applications including environmental monitoring for agricultural and ecological purposes, wild life monitoring, remote patient monitoring in electronic health care systems, building automation, and reconnaissance applications for military purposes. Sensor networks typically consist of a large number of sensor nodes and a few base stations. The sensor nodes measure some physical phenomena (e.g., temperature, humidity, vibration) that are important in the given application, and report their sensor readings to the base stations (typically via wireless communication channels). As both the number of the sensors and the amount of the measurements that they perform can be large, in many applications, the base stations aggregate the individual sensor readings into a compact report. Aggregation can be useful to keep the amount of information that need to be handled under control, and to improve the energy-efficiency of the network. The typical aggregation functions include the average, the minimum, and the maximum.

A potential problem is that sensor readings can be compromised before they reach the base station. This can be achieved by an attacker either by modifying the content of the data packets that carry the sensor readings, or by altering the environmental parameters around some sensors and corrupt their readings. While the former type of attack can be detected by standard cryptographic message authentication and integrity protection techniques, the latter type of attack cannot be detected, nor prevented, by cryptographic means. In addition, the latter type of attack is relatively easy to carry out: Firstly, an attacker can easily approach a sensor node, as sensor networks 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,

* Corresponding author. Tel.: +36 203373064.
E-mail addresses: buttyan@crysys.hu (L. Buttyán), schaffer@crysys.hu (P. Schaffer), vajda@crysys.hu (I. Vajda).

1570-8705/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2008.09.006

a pocket lamp, or a glass of water can be used to corrupt temperature, light, and humidity measurements, respectively). Unfortunately, many useful aggregation functions (including those mentioned above) 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.

The goal of resilient data aggregation is to alleviate the problem described above. More specifically, resilient data aggregation schemes try to minimize the effect of an environment altering attacker on the output of the aggregation function. However, the related solutions often make the simplifying assumption that the sensor nodes produce *independent* and identically distributed measurements. In reality, however, the measurements made by the sensors always have some kind of relationship among them. This relationship can be either temporal correlation (i.e., when the nodes' sensing results show regularity in time), or spatial correlation (i.e., when the nodes' physical proximity is the basis of the relationship).

Contrary to several prior work on resilient data aggregation in sensor networks, in this paper, we assume that the sensor readings are *correlated*. In particular, we will focus on spatially correlated measurements. The rationale is that, in most of the sensor network applications, one needs to have a densely deployed network in order to satisfy the sensing coverage and radio connectivity requirements. Consequently, sensors in proximity will measure spatially correlated values of the same phenomenon where the degree of correlation increases as the internode distance decreases.

Spatial correlation can be exploited to cross-check the sensor readings, testing whether there is an (environment altering) attack or not. This naturally existing characteristic of the sample produced by the sensor network helps in improving the attack detection algorithms proposed so far in this context. Furthermore, considering correlation is a significant step towards having a more realistic data model of sensor networks in general.

Hereinafter, we introduce our sensor network model that is able to handle spatial correlation, and we also introduce a novel resilient data aggregation scheme developed for sensor networks that exploits the spatial correlation of the sensor readings. Moreover, we study our proposed data aggregation scheme analytically and by means of simulation, and show how the effectiveness of attack detection can be improved by considering correlation. Our previously published short conference paper [20] deals with the same problem. This paper should be viewed as a follow-up and substantially extended version of that short paper.

The rest of this paper is organized as follows: In Section 2, we summarize the papers considering correlation in sensor networks and the papers related to resilient aggregation. In Section 3, we present our sensor network model. In Section 4, we introduce our novel correlation-based resilient data aggregation approach, and we evaluate its efficiency. Then, in Section 5, we answer some emerging questions, and finally, in Section 6, we conclude our work and propose some interesting future research topics.

2. Related work

Even though the naturally existing phenomenon of correlation is sometimes neglected in research papers considering sensor measurement data, it can be exploited in many ways. In the following, we present the related papers in the field of data aggregation, deviation detection and attack detection considering correlation.

A research paper that aims at data aggregation considering correlation is [30], the authors of which propose an aggregator node election mechanism that aims at load balancing too. According to this mechanism, the network is partitioned into equally sized sectors, wherein the aggregator nodes – that are selected considering the correlation – collect the data from their children in case an event occurs. In [29], correlation is exploited in in-network aggregation. The highly correlated nodes are assumed to have similar measurement results, therefore, only one of them is sufficient to fulfill the sensing task. Relying on this assumption, the proposed solution reduces the number of transmissions and provides approximate results to aggregate queries by utilizing the spatial correlation of sensor data.

There are papers that aim at detecting anomalies (outliers, deviations) in the system usually by exploiting the phenomenon of correlation, but not in the context of sensor networks. In [14], the authors propose a method to detect anomalous network conditions with the help of PCA (principal component analysis), while in [17], one can read about an outlier detection scheme that uses approximate computations in order to accelerate the operation, and detects outliers with the help of the so called 'multi-granularity deviation factor'. While these papers are not designed for sensor networks, there is a related solution for sensor networks as well [16], in which the authors deal with the problem of identification of deviating values in streaming data. Regrettably, this latter paper assumes a special network topology with a powerful backbone, and applies kernel density estimators, thus restricting itself to i.i.d. samples.

These papers do not consider attacks, only anomalies (or outliers, deviations). The main difference between the two concepts is that anomalies are random events, while attacks are controlled events that aim at disturbing some functionalities of the sensor network. The problem of defending such attacks in sensor networks is obviously important, hence, there are more and more papers discussing countermeasures.

An example of such papers is [22], in which the authors propose a method to reduce the effect of unauthorized data inserted by sybil and compromised nodes. The paper exploits correlation using a modified, sliding-window t-test that will point out the nodes that are suspected to be captured or sybil nodes, and these nodes have to authenticate themselves in this case. If a node fails to authenticate itself, its message will be dropped and the malfunctioning will be reported to the base station. Another related paper is [28], in which one can read about an en-route filtering method against injected false messages using multiple MACs (message authentication codes). The main idea here is that the sensor nodes have

disjoint sets of cryptographic keys, and the nodes that sense the same event can attach multiple MACs calculated by each of them to the message. Thus, if an attacker wants to forge a message, it has to forge several MACs as well, but this forged message will be detected during its route to the base station as the forwarding nodes can check the validity of the MACs with some probability. Regrettably, neither [22] nor [28] propose resilient solutions: these solutions are not applicable against an outsider attacker who alters the measured parameters of the environment in order to have the sensors perform falsified measurements. This kind of attack results in messages that are cryptographically sound but false in content. This is a serious security threat in the sensor networks, as already mentioned in Section 1.

Some researchers already considered the problem of such messages that cannot be filtered out using cryptographic checks, however, usually under the assumption that the measurements of the sensors are independent. One of the first research papers on this topic was [26]. The author investigates the resilience of the commonly known aggregation functions like, for example, the average, the min/max, and the median. Not surprisingly, most of these function are not resilient even against one compromised sample element (which can originate from only one compromised node), and only the median is declared to be resilient. A question naturally arises: what can we do if we do not want to calculate the median of the sample, but something else in a secure way? In [9] the authors address the same problem of compromised sensor readings. However, this paper shows a method only for attack detection. In the model of [9], the attacker does not only want to cause a distortion in the output of the aggregation function, but he also wants to remain undetected. This trade-off helps in upper bounding the strength of the attacker notably. Another paper of the same authors [8] gives a complex answer to this question by introducing the RANBAR algorithm. This algorithm is able to do statistical sample filtering, thus, it helps to obtain a cleaned sample which can be a basis for secure calculation of any kind of aggregates, even those that were declared to be not resilient in [26].

In the most related paper [10], one can read about a security solution that already assumes compromised nodes and the defense against them with the help of correlation. The authors employ PCA (principal component analysis) in order to detect the misbehaviour of the nodes and filter out their measurements. According to the simulation results, the proposed methodology overperforms conventional anomaly detection approaches. However, the paper assumes a special network topology with more powerful primary nodes that, at the same time, cannot be compromised. Moreover, the a priori assumption in PCA is that the most important components (i.e., sensors) are those that have a high variance in their values, which is not true in general in our case.

After having presented the related literature, in the following sections we introduce our solution for resilient data aggregation in sensor networks. Our approach, described in detail in Sections 3 and 4, exploits correlation to ensure the resilience of data aggregation even in case of an attacker's activity.

3. General assumptions

In the next subsection, we present the set of assumptions we made on the attacker, which together is called attacker model in the security literature. After that, we present the data model that we employ for the calculations throughout the paper.

3.1. Assumptions on the adversary

The adversary we consider is able to produce some kind of "offsets" which are added to the measurements of the sensors. These offsets are under the control of the adversary, but are considered to be independent and identically distributed. Moreover, those are considered to be of the same kind as the sensor readings, e.g., temperature in case of thermometer sensors or light in case of photometer sensors. This attack can totally distort the aggregate considering the commonly used aggregation functions like the average or the min/max.

We do not restrict the adversary in the number of sample elements he is able to compromise, but we assume that the adversary's knowledge do not extend to the distribution of the sample produced by the sensor network, neither to the size of the sample gathered by the base station in a given query (some of these assumptions will be relaxed later in Section 4.2.3). Finally, we do not consider any particular distribution for the attacker's offset.

An example of such an attack is the following. Let us assume that we have a sensor network on the vineyard that measures some microclimate characteristics by calculating the min/max (or the average, etc.) of the measurements of the individual nodes [2,7,6,23]. The owner of the vineyard is assisted by the aggregated reports in the decision making about what task is needed to be done on the vineyard, which ensures the maximum quality of the grapes. Obviously, a malicious outsider can easily mislead the aggregate by approaching only one sensor node and compromising its measurement for example by a lighter, or by using chemicals, according to the measured characteristics. The misleaded aggregate can encourage the owner to perform inappropriate operations (e.g., grape harvesting in wrong time, inappropriate usage of chemicals, etc).

Another example can be considered in the case of bridge monitoring sensor networks that are deployed to permanently monitor the structural and seismic conditions of the bridge [15,4]. Even one compromised measurement in the aggregate of these measurements can cause false alarms for the bridge maintainers, and what is more, suppressed alarms can lead to disasters because of the missing maintenance.

We emphasize that the mentioned attacker does not have to tamper with the nodes or reverse-engineer the crypthographic keys, neither needs he to destroy the communication protocols used in the network – he only needs physical proximity!

3.2. The data model

In our envisioned application, the base station collects a sample of measurements from the sensors and tries to aggregate them in a secure way. Each sensor contributes to this sample with its measurement by replying to the base station's query in an encrypted message. (We note that assuming even public key encryption in sensor networks is not far-fetched according to [18]. Moreover, we note that our scheme supports distributed in-network aggregation as well, see Section 5 for the related discussion.) Upon reception of the messages the base station decrypts the messages and aggregates their information content. The aggregation is done in two steps: At first, the sample is analyzed and a decision is made whether it is compromised or not. After that, an aggregation step is performed depending on the previous decision. The aggregation step is different for the two outputs of the decision function, namely when an attack is detected or when no attack is detected. If there is no attack detected then usual aggregation is performed, otherwise the final output is calculated by extrapolation based on the previous outputs (see Fig. 1). This separation of cases helps us to obtain a significantly smaller distortion at the output of the aggregation function than having done the aggregation without attack detection.

We assume that the sensor network data is normally distributed. The choice of the normal distribution is a common assumption in practice when measurement data is considered. However, we note that the algorithms we propose in the following sections are applicable to any kind of sampling distributions; the assumption on the normal distribution is used only in the derivation of the analytical and simulation results in this paper.

In order to be able to measure the gain of our approach, we model the sensor network to produce measurements that can be represented by identically distributed random variables, but instead of assuming the independence of these random variables we exploit the correlation among them (in other words, we consider dependent random variables). Therefore, our sensor network data model consists of the following elements:

- n: number of sensor readings in the sample.
- t: number of readings compromised by the attacker.
- X_i: normally distributed random variable denoting the ith uncompromised reading ($X_i \sim \mathcal{N}(\mu, \sigma)$, $0 < i \leqslant n$).
- $r_{X_i X_j} = r$: correlation coefficient between X_i and X_j, $\forall i, j$, $i \neq j$.
- G_i: arbitrarily distributed random variable denoting the additive offset produced by the attacker (G_i is independent of X_i, $\forall i$).
- $Z_i = X_i + G_i$: random variable denoting the compromised sample elements ($0 < i \leqslant t$).

As this model handles the dependence of the sensor measurements, it can help us to quantify the power of correlation in attack detection. In the next section, we will show how this quantification can be performed.

4. Exploiting correlation in resilient data aggregation

Correlation among sample elements is a naturally existing phenomenon. In this section, we show how this correlation can be exploited. We start with a simplified scenario of two nodes in Section 4.1 in order to get a first insight into the problem and to prepare the ground for the general case. After that, we generalize our model in Section 4.2 for arbitrary number of nodes and attack strengths.

4.1. The two-nodes scenario

As a first step, we investigate the case when there are only two sample elements (i.e., $n = 2$), and there is at most one element that is attacked (i.e., $t \leqslant 1$). Our primary aim now is to pursue attack detection on this 2-element sample with a small error probability (both false negative and false positive). Then, based on this decision, we are able to perform data aggregation of the same 2-element sample with a remarkably lowered distortion, where the distortion is defined as the expected value of the squared absolute difference between the aggregate of the sample and the aggregate in case there is no attack. Our secondary aim is to show how correlation influences our results calculated for the distortion.

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. $p_{X_1|X_2}(\cdot|x')$
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**

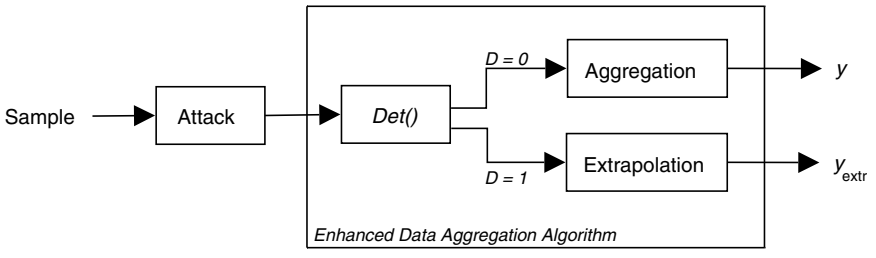


Fig. 1. Resilient aggregation scenario including the attacker and the data processing part.

The solution we propose to fulfill our primary aim is the Attack Detection Algorithm $Det(x_1, x_2)$ (Algorithm 1). This algorithm randomly chooses one of the two elements from the sample and computes the $(1-\alpha)\%$ confidence interval for the remaining one conditioned on the chosen one, where α is the false positive probability. If the remaining one is inside this confidence interval, then the output of the algorithm is that there is probably no attack ($D = 0$), otherwise the algorithm signals that an attack is detected ($D = 1$).

This straightforward approach already exploits correlation by using the conditional probability density function $p_{X_1|X_2}(\cdot|\cdot)$ which is assumed to be known. In most of the cases this can be a realistic assumption as the base station can perform data gathering and can establish an estimation of $p_{X_1|X_2}(\cdot|\cdot)$ just after the deployment of the sensor network when the probability of being already attacked is small. We note, however, that the knowledge of $p_{X_1|X_2}(\cdot|\cdot)$ 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, we do not assume any a priori knowledge about the expected value of the measurement data.

The output of the Attack Detection Algorithm can be applied in selecting the adequate way of data aggregation. If no attack is indicated then the sample can be handled in the usual way, e.g., its average can be calculated without the fear of obtaining a highly distorted aggregate. Otherwise, equipped with the knowledge that the sample is compromised with high probability, one can mitigate the effects of an attacker by handling the sample in a special way. Usually, dropping the compromised sample is the easiest method to apply, while extrapolating the current aggregate from the previous (unattacked) results can guarantee a small distortion without relying on other information. The type of the extrapolation can be suitably chosen to the characteristics of the data one is going to measure.

This 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. For example, y_{extr} can be the output of the last run of the data aggregation algorithm when the attack detection algorithm detected no attack; this possesses the smallest distortion for ordinary samples.

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
3: Output $= y_{\text{extr}}$
4: **else**
5: Output $= y$
6: **end if**

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

4.1.1. Why not using standard statistical decisions instead of $Det(\cdot,\cdot)$?

Decision theory is a well-elaborated part of statistics. It is concerned with the topic of how to behave optimally under uncertainty. The basic guideline in decision theory is minimizing the expected loss encountered after the decision. Generally speaking, we have the same objective in this paper: we want to minimize the distortion of the aggregation function. Thus, the distortion can be considered as the loss in our case, while the decision we have to make is about signalling an attack or not. Why not using then well-known statistical decisions instead of inventing a new one? To answer this question we have to take a deeper look at the *modus operandi* of the decision algorithms proposed so far. The two most prevalent statistical decisions we investigate are the Bayesian decision and the Maximum Likelihood decision.

Informally, the Bayesian decision is concerned with making a decision about the state of nature based on how probable that state is. Therefore, Bayesian decision theory plays a role when there is some *a priori* information about the states we are trying to classify. As we want to decide whether there is an attack occurred or not, the a priori information would be in our case the probability of facing an attack. However, our attacker model presented in Section 3.1 does not contain any kind of information about this probability. In other words, we do not rely on assumptions about the attacker's attacking frequency or distribution in time. Therefore, the Bayesian decision that requires information about the attacking probability cannot be applied in our case.

The Maximum Likelihood decision seems to be more attractive in the scenario proposed in this paper. Generally, the Maximum Likelihood approach decides to that state of nature for which it holds that the value of the p.d.f. for the input conditioned on that state is the maximum value among all the values of p.d.f's conditioned on other states for the same input. The sample received from the sensor nodes can be considered as the mentioned input, while the states of nature are 'attack' or 'no attack'. The problem with this approach is that without assuming a concrete distribution of the attacker's additive offset we cannot figure out the p.d.f. of a vector of sample elements conditioned on the class 'attack'. Therefore, regrettably, the Maximum Likelihood decision needs too much information that is not available in our model and thus, it is not applicable either in our case.

4.1.2. Evaluation of the enhanced data aggregation algorithm under a gaussian data model

To quantify the gain in the distortion of the output of the Enhanced Data Aggregation Algorithm, we first have to evaluate the error probabilities of the Attack Detection Algorithm. These probabilities are the false positive (α) and the false negative (β) probabilities. α is the probability

of signalling an attack in the unattacked case, while β is the probability of not signalling the attack in the attacked case. In order to be able to define β, we fix α to 0.1 (i.e., we tolerate 10% of false alarms). Moreover, for the evaluation we assume that the distribution of G_i is the Gaussian distribution with parameters $\tilde{\mu}$ and $\tilde{\sigma}$ (i.e., $G_i \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma})$). (We note that this assumption is only needed for the calculations below, Algorithms 1 and 2 do not rely on it. We also note that a more general attacker will be considered later and analyzed by means of simulation in Section 4.2.3.) Here, the choice of the Gaussian distribution simplifies the analysis and its two parameters allow us to consider attacks of significantly different styles. Without loss of generality, we further assume that the first sample element is compromised, i.e., $Z_1 = X_1 + G_1$. As $t = 1$, we can set aside the lower indices of the symbols corresponding to the attacker, thus $Z = X_1 + G$. Based on these, the β error probability can be determined by averaging the two particular false negative error probabilities corresponding to the two cases when (i) we select the compromised element as the condition (i.e., $x' = z$) or (ii) we select the uncompromised reading for the same role (i.e., $x' = x_2$). The averaging is justified by the fact that both of these events have a probability of 0.5 to occur because of the randomness of the selection. Formally,

$$\beta = \frac{1}{2}(\beta^{(1)} + \beta^{(2)}), \tag{1}$$

where

$$\beta^{(1)} = \int_{-\infty}^{\infty} \left[\int_{b_1(z)}^{b_2(z)} p_{X_2|Z}(u|v)\mathrm{d}u \right] p_Z(v)\mathrm{d}v \tag{2}$$

$$= \int_{-\infty}^{\infty} \int_{b_1(z)}^{b_2(z)} p_{X_2,Z}(u,v)\mathrm{d}u\mathrm{d}v \tag{3}$$

$$\beta^{(2)} = \int_{-\infty}^{\infty} \left[\int_{b_1(x_2)}^{b_2(x_2)} p_{Z|X_2}(u|v)\mathrm{d}u \right] p_{X_2}(v)\mathrm{d}v \tag{4}$$

$$= \int_{-\infty}^{\infty} \int_{b_1(x_2)}^{b_2(x_2)} p_{Z,X_2}(u,v)\mathrm{d}u\mathrm{d}v. \tag{5}$$

The $b_1(z)$, $b_2(z)$, $b_1(x_2)$ and $b_2(x_2)$ integration bounds are defined with the help of the previously fixed false positive probability as

$$\int_{-\infty}^{b_1(z)} p_{X_1|X_2}(u|z)\mathrm{d}u = \frac{\alpha}{2} \tag{6}$$

$$\int_{b_2(z)}^{\infty} p_{X_1|X_2}(u|z)\mathrm{d}u = \frac{\alpha}{2} \tag{7}$$

$$\int_{-\infty}^{b_1(x_2)} p_{X_1|X_2}(u|x_2)\mathrm{d}u = \frac{\alpha}{2} \tag{8}$$

$$\int_{b_2(x_2)}^{\infty} p_{X_1|X_2}(u|x_2)\mathrm{d}u = \frac{\alpha}{2}, \tag{9}$$

respectively. Additionally, the correlation coefficient in $p_{X_2,Z}(\cdot,\cdot)$ is calculated as

$$r_{X_2,Z} = \frac{E[(X_2 - \mu)(X_1 + G - \mu - \tilde{\mu})]}{\sigma\sqrt{\sigma^2 + \tilde{\sigma}^2}} \tag{10}$$

$$= r_{X_1,X_2} \frac{\sigma}{\sqrt{\sigma^2 + \tilde{\sigma}^2}}, \tag{11}$$

and the correlation coefficient in $p_{Z,X_2}(\cdot,\cdot)$ is $r_{Z,X_2} = r_{X_2,Z}$.

With the help of β, we can analyze our Enhanced Data Aggregation Algorithm from its distortion's point of view. As the most interesting aggregation function is the average because of its vulnerability (only one compromised measurement can totally mislead it) and its widespread usage, we considered it in our analysis too. To evaluate the distortion of the output of Algorithm 2, we have to distinguish two basic cases: the case when an attack happens, and another one when there is no attack. We introduce the following notations:

- A: indicator random variable denoting whether there is an attack or not (0 – no attack, 1 – attack)
- Y: random variable denoting the average of the sample
- Y_{extr}: random variable denoting the minimum distortion output in case an attack is detected
- $\widehat{Y}$: random variable denoting the average of the sample elements when there is no attack

Considering the first reading to be compromised (without loss of generality), the distortion in the first case can be expressed as

$$\mathrm{d}(Y|A=1) = E[|Y - \widehat{Y}|^2 | A = 1] \tag{12}$$

$$= E[|Y - \widehat{Y}|^2 | A = 1, D = 1] \cdot (1 - \beta) \tag{13}$$

$$+ E[|Y - \widehat{Y}|^2 | A = 1, D = 0] \cdot \beta \tag{14}$$

$$= E|Y_{\text{extr}} - \widehat{Y}|^2 \cdot (1 - \beta) + \frac{1}{4}(\tilde{\mu}^2 + \tilde{\sigma}^2) \cdot \beta. \tag{15}$$

While in the second case the distortion can be formalized as

$$\mathrm{d}(Y|A=0) = E[|Y - \widehat{Y}|^2 | A = 0] \tag{16}$$

$$= E[|Y - \widehat{Y}|^2 | A = 0, D = 1] \cdot \alpha \tag{17}$$

$$+ E[|Y - \widehat{Y}|^2 | A = 0, D = 0] \cdot (1 - \alpha) \tag{18}$$

$$= E|Y_{\text{extr}} - \widehat{Y}|^2 \cdot \alpha. \tag{19}$$

To show how much gain our Enhanced Data Aggregation Algorithm induces compared to a scenario where no attack detection is employed, we define d_{imp} as the improvement in the distortion in case of an attack as follows:

$$d_{\text{imp}} = d(Y|A = 1, D = 0) - d(Y|A = 1) \tag{20}$$

$$= \frac{1}{4}(\tilde{\mu}^2 + \tilde{\sigma}^2) - [E|Y_{\text{extr}} - \widehat{Y}|^2 \cdot (1 - \beta) + \frac{1}{4}(\tilde{\mu}^2 + \tilde{\sigma}^2)\beta]$$

$$\cong \frac{1}{4}(\tilde{\mu}^2 + \tilde{\sigma}^2) \cdot (1 - \beta), \tag{21}$$

where we assume that $E|Y_{\text{extr}} - \widehat{Y}|^2$ is close to zero. In Fig. 2, one can see a plot of d_{imp} where the different curves belong to different correlation coefficients. The horizontal axis corresponds to the expected value of the attacker's distribution (i.e., $\tilde{\mu}$). For the calculations we choose $\mu = 0$, $\sigma = 1$, and $\tilde{\sigma} = 1$. We note that the choice of $\tilde{\sigma}$ in the range $[0.5, 1.5]$ does not alter the results significantly. In the figure, the steeply ascending lines show that the improvement in the distortion grows with a growing difference between μ and $\tilde{\mu}$. The fact the curve of $r = 0.5$ runs near to the curve of $r = 0.95$ clearly indicates that our

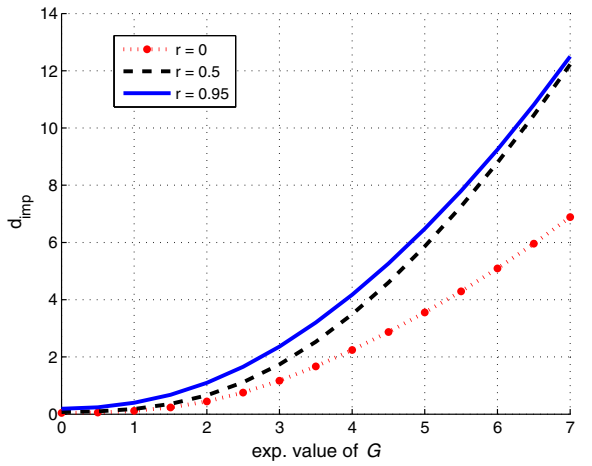


Fig. 2. Calculated values for d_{imp} for different values of the correlation coefficient r.

approach considerably exploits even correlations of moderate power.

As a second comparison, we show how much influence the correlation has on the distortion. In Fig. 3, one can see the distortion $\mathrm{d}(Y|A=1)$ for different values of the correlation coefficient r. The horizontal axis represents the expected value of the attacker's distribution (i.e., $\tilde{\mu}$). The corresponding values for the calculations are $\mu=0$, $\sigma=1$, and $\tilde{\sigma}=1$. Here again, assuming that $E|Y_{\text{extr}}-\widehat{Y}|^2$ is close to zero, we can characterize the distortion as

$$\mathrm{d}(Y|A=1) \cong \frac{1}{4}(\tilde{\mu}^2+\tilde{\sigma}^2)\cdot\beta. \tag{22}$$

In Fig. 3, the difference in the form of the curves for the dependent cases ($r>0$) and the independent case ($r=0$) shows that considering correlation helps in maintaining a very moderate distortion in the aggregate in case of an attack. When the sample elements are independent, the distortion caused by the adversary grows steeply with $\tilde{\mu}$,

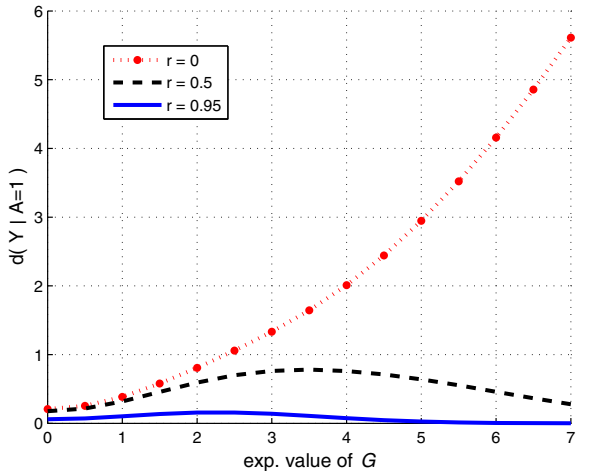


Fig. 3. Distortion caused by the adversary for different values of the correlation coefficient r.

while in the dependent cases the effects of an attack are strictly upper bounded, even when the correlation is moderate.

To understand the behaviour of Algorithm 1 more deeply, we compared it to the already detailed Maximum Likelihood decision. As mentioned in Section 4.1.1, the Maximum Likelihood decision is not applicable in our data and attacker model, however, its importance in decision theory lead us to compare its efficiency to the efficiency of Algorithm 1 in a significantly restricted model. The restriction is the following: we assume that the attacker's distribution is *a priori* known. We 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 our approach; the Attack Detection Algorithm does not need to know the attacker's distribution while the Maximum Likelihood decision needs. For the sake of simplicity, we assume that the attacker's distribution is the Gaussian distribution with known expected value and variance.

The Maximum Likelihood decision is the following. Let us take the joint p.d.f. of the sample in case there is no attack (i.e., $p_{X_1X_2}$) and divide it with the joint p.d.f. corresponding to the attacked case (i.e., p_{X_1Z} or p_{ZX_2}). An attack is signalled if this quotient is smaller than T. More formally, $D=1$ if

$$\frac{p_{X_1X_2}(\boldsymbol{x})}{\frac{1}{2}p_{X_1Z}(\boldsymbol{x})+\frac{1}{2}p_{ZX_2}(\boldsymbol{x})} < T, \tag{23}$$

where T can be obtained with the help of the false positive probability α. Therefore, T can be determined using that

$$\alpha = \int_{\mathscr{R}} p_{X_1X_2}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{24}$$

where $\mathscr{R}$ is defined as

$$\mathscr{R} = \left\{\boldsymbol{x} : p_{X_1X_2}(\boldsymbol{x}) < T\left(\frac{1}{2}p_{X_1Z}(\boldsymbol{x})+\frac{1}{2}p_{ZX_2}(\boldsymbol{x})\right)\right\}, \tag{25}$$

After having the Maximum Likelihood decision described, we have to evaluate its probability of missed detection. This can be formalized as

$$\beta = 1-\left(\frac{1}{2}\int_{\mathscr{R}} p_{X_1Z}(\boldsymbol{x})+\frac{1}{2}\int_{\mathscr{R}}\frac{1}{2}p_{ZX_2}(\boldsymbol{x})\right). \tag{26}$$

To be able to observe the effect of the Maximum Likelihood decision on the distortion, we have put it in the Enhanced Data Aggregation Algorithm in place of $Det(\cdot,\cdot)$. Using the new values of β, the improvement in the distortion of the Enhanced Data Aggregation Algorithm can be calculated using Eq. (21).

Fig. 4 shows the results of the comparison of the Attack Detection Algorithm and the Maximum Likelihood decision, both as a building block in the Enhanced Data Aggregation Algorithm. The corresponding values for the calculations are $\mu=0$, $\sigma=1$, $\tilde{\sigma}=1$. As one can see from Fig. 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

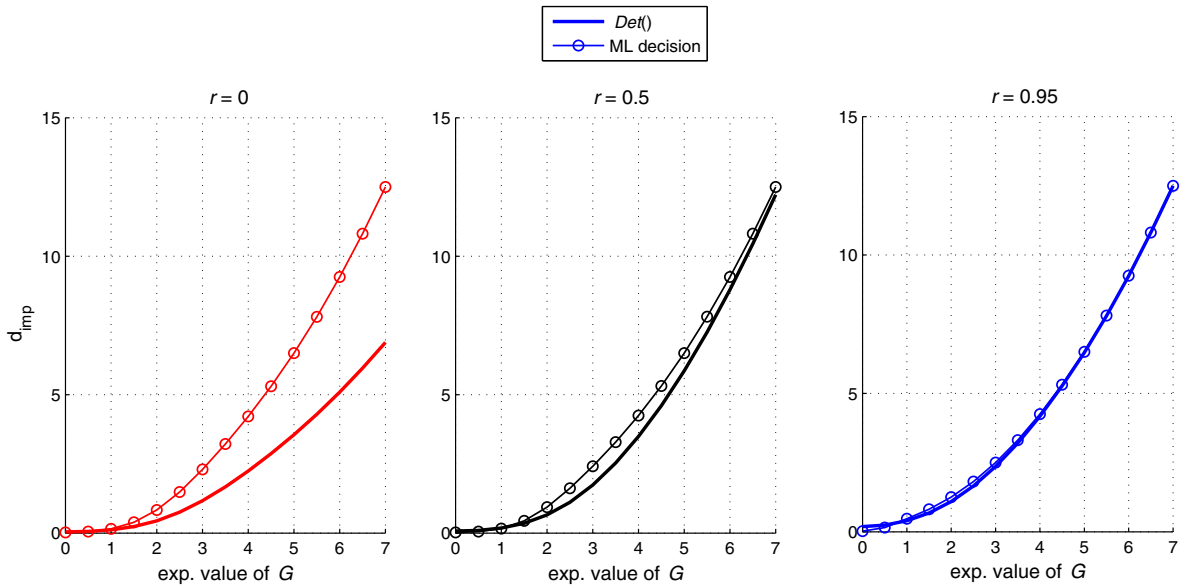


Fig. 4. Comparison of Maximum Likelihood decision and the Attack Detection Algorithm.

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. Nevertheless, we emphasize again that the Maximum Likelihood approach is only applicable if one knows the distribution of the attacker's offset, while the Attack Detection Algorithm does not need this knowledge.

Figs. 2–4 clearly show that correlation has a significant influence on the attack detection capabilities of Algorithm 1 and therefore on the distortion that the attacker is able to cause in the output of Algorithm 2. Compared to the independent case (i.e., when $r = 0$), considering the naturally existing correlation between the sample elements results in smaller distortion and allows the base station to make nearly as precise decisions as for example the Maximum Likelihood approach which needs more knowledge about the attacker in order to be able to operate. In other words, the attacker's abilities are more restricted when the base station maintains a correlation-based data model.

Using the preliminary data model consisting of only two nodes from which one is possibly attacked we are able to quantify the "strength" of correlation. The results justify our suspicion: exploiting correlation can help in developing data aggregation algorithms for sensor networks that are more powerful from the resilience point of view than algorithms not considering correlation. Now that the importance of correlation is clarified, we can go further by enabling our algorithms to elaborate on data sets that are containing more than two elements. In Section 4.2 we will show how this generalization can be performed.

4.2. Generalization using sample halving

Usually, sensor networks are imagined to contain a high number of sensor nodes, and in our simplified case the number of nodes is strictly related to the sample size. Thus, in this subsection we propose a generalized approach for attack detection and resilient aggregation in sensor networks that is able to handle a sample of arbitrary size. That means that in this subsection we consider samples for which

- $n \geqslant 2$
- $t \geqslant 1$, i.e., the attacker's strength is also considered to be arbitrary.

As the Attack Detection Algorithm and the Enhanced Data Aggregation Algorithm are efficient considering a small sample, it is a natural idea to reuse them in this general case. In the first step, one has to shrink a sample of n elements into a sample of two elements which can be achieved, for example, by halving the sample into two partitions and compressing the partitions into one element each. The halving is done in a random way, i.e., each element has a 50% chance to get into the first partition and the same holds for the second partition too. The compression can be done for the two partitions independently from each other by e.g. averaging the halves. In our case, the partitions do not need to have equal size but for simplicity we require this property now. With this sample halving approach we are able to reduce the general problem (i.e., $n \geqslant 2$) to a special case (i.e., $n = 2$) where we can apply our previously introduced Attack Detection Algorithm and Enhanced Data Aggregation Algorithm.

A sketch of the sample halving approach can be seen in Fig. 5, in which a sample with six elements is represented

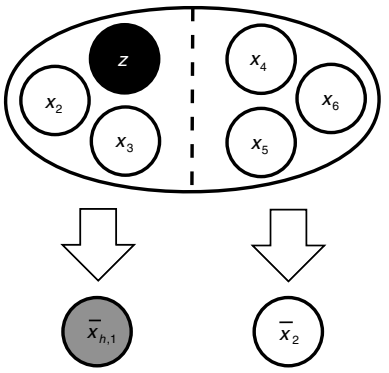


Fig. 5. A sketch of the sample halving approach.

by circles, where the white circles correspond to ordinary elements and the black circle corresponds to an element that is compromised by the adversary. The sample halving approach divides the sample into two partitions in a random way and compresses the two partitions independently from each other to obtain a sample of size two. As the first partition contains a compromised element its compressed counterpart is also considered as compromised, but since the averaging blurs the effect of the adversary the circle of the resultant value is grey instead of black.

To be able to use the Attack Detection Algorithm $Det(\cdot,\cdot)$ and the Enhanced Data Aggregation Algorithm presented in Section 4.1, we have to obtain the conditional p.d.f. of the average of the first partition conditioned on the average of the second partition, as instead of $p_{X_1|X_2}(\cdot|\cdot)$ we need $p_{\overline{X}_1|\overline{X}_2}(\cdot|\cdot)$ in Algorithm 1 to evaluate the corresponding confidence interval in the general case. Again, this can be obtained by performing measurements just after the deployment of the sensor network when the probability of being already attacked is small. We note that the knowledge of $p_{\overline{X}_1|\overline{X}_2}(\cdot|\cdot)$ does not assume anything about the knowledge of the sampling distribution of the measured parameter of the sensor network.

With this modified assumption we can reduce the problem of resilient data aggregation on an arbitrary-sized sample to the already solved problem of data aggregation on a sample of size two. Therefore, we are now able to perform attack detection and resilient aggregation on a sample without restriction on its size or the number of compromised elements. In the next subsection, we formally evaluate the sample halving approach.

4.2.1. Evaluation of the generalized algorithm under a gaussian data model

The quantification of the gain introduced by the Enhanced Data Aggregation Algorithm in the case of samples of arbitrary size is similar to the evaluation of the case of 2-element samples in Section 4.1.2. However, even if some of the formulas look similar, the reason of their usage can be very different compared to the previous case. Moreover, the increased number of possibly compromised elements renders the analysis a bit more difficult.

Firstly, we introduce the notations needed:

- $\mathbf{X}$: normally distributed random vector denoting the original sample ($\mathbf{X} \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$).
- $\mathbf{X}_h$: arbitrarily distributed random vector denoting the sample in case of an attack ($\mathbf{X}_h \sim \mathcal{N}_n(\mu_h, \boldsymbol{\Sigma}_h)$).
- $\overline{\mathbf{X}}$: normally distributed random vector produced by averaging the halves of $\mathbf{X}$ in the unattacked case ($\overline{\mathbf{X}} \sim \mathcal{N}_2(\overline{\boldsymbol{\mu}}, \overline{\boldsymbol{\Sigma}})$).
- $\overline{\mathbf{X}}_h$: random vector produced by averaging the halves of $\mathbf{X}_h$ in case of an attack ($\overline{\mathbf{X}}_h \sim \mathcal{N}_2(\overline{\boldsymbol{\mu}}_h, \overline{\boldsymbol{\Sigma}}_h)$).
- $r_{\overline{X}_1\overline{X}_2}$: correlation coefficient between the elements of $\overline{\mathbf{X}}$.

The sample $\mathbf{X}$ in the unattacked case has a multivariate normal distribution with mean (expected value) vector

$$\boldsymbol{\mu} = (\mu, \ldots, \mu)^{\mathrm{T}}, \tag{27}$$

and with covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma^2 & r\sigma^2 & \ldots & r\sigma^2 \\ r\sigma^2 & \sigma^2 & \ldots & r\sigma^2 \\ \vdots & \vdots & \ddots & \vdots \\ r\sigma^2 & r\sigma^2 & \ldots & \sigma^2 \end{pmatrix} \tag{28}$$

In case an attack happens, the mean vector and the covariance matrix of the compromised sample $\mathbf{X}_h$ are respectively (without loss of generality)

$$\boldsymbol{\mu}_h = \boldsymbol{\mu} + \boldsymbol{\mu}_\Delta, \tag{29}$$

and

$$\boldsymbol{\Sigma}_h = \boldsymbol{\Sigma} + \boldsymbol{\Sigma}_\Delta, \tag{30}$$

where $\boldsymbol{\mu}_\Delta$ is a column vector the first t elements of which are $\tilde{\mu}$'s, and $\boldsymbol{\Sigma}_\Delta$ is a $n \times n$ matrix containing only zero elements except from its first diagonal where the first t elements are $\tilde{\sigma}^2$.

The averaging transformation of the partitions can be described by matrix $\mathbf{M}$ which is a $2 \times n$ matrix with the following entries:

$$\mathbf{M} = \begin{pmatrix} \frac{2}{n} & \ldots & \frac{2}{n} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \frac{2}{n} & \ldots & \frac{2}{n} \end{pmatrix}. \tag{31}$$

In this generalized case again, the first step is to formalize the false negative probability β as

$$\beta^{(t_1,t_2)} = \frac{1}{2}(\beta^{(1)} + \beta^{(2)}), \tag{32}$$

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

$$\beta^{(1)} = \int_{-\infty}^{\infty} \int_{b_1(\bar{x}_{h,1})}^{b_2(\bar{x}_{h,1})} p_{\overline{X}_{h,2},\overline{X}_{h,1}}(u,v)\mathrm{d}u\mathrm{d}v \tag{33}$$

$$\beta^{(2)} = \int_{-\infty}^{\infty} \int_{b_1(\bar{x}_{h,2})}^{b_2(\bar{x}_{h,2})} p_{\overline{X}_{h,1},\overline{X}_{h,2}}(u,v)\mathrm{d}u\mathrm{d}v, \tag{34}$$

similarly to the definitions in Section 4.1.2. The related distributions can be defined with the help of matrix multiplications $\mathbf{M}\boldsymbol{\mu}_h$ and the $\mathbf{M}\boldsymbol{\Sigma}_h\mathbf{M}^{\mathrm{T}}$ which result respectively in

$$\overline{\boldsymbol{\mu}}_h = \begin{pmatrix} \mu + \frac{2}{n} t_1 \tilde{\mu} \\ \mu + \frac{2}{n} t_2 \tilde{\mu} \end{pmatrix}, \tag{35}$$

and

$$\overline{\boldsymbol{\Sigma}}_h = \begin{pmatrix} \overline{\boldsymbol{\Sigma}}_{h,11} & r\sigma^2 \\ r\sigma^2 & \overline{\boldsymbol{\Sigma}}_{h,22} \end{pmatrix}, \tag{36}$$

where

$$\overline{\boldsymbol{\Sigma}}_{h,11} = \left(\frac{2}{n}\right)^2 t_1 \tilde{\sigma}^2 + \frac{2}{n}\sigma^2 + \left(1 - \frac{2}{n}\right) r\sigma^2. \tag{37}$$

$$\overline{\boldsymbol{\Sigma}}_{h,22} = \left(\frac{2}{n}\right)^2 t_2 \tilde{\sigma}^2 + \frac{2}{n}\sigma^2 + \left(1 - \frac{2}{n}\right) r\sigma^2. \tag{38}$$

Based on these, the distribution of $\overline{X}_{h,1}$ is

$$\overline{X}_{h,1} \sim \mathcal{N}\left(\overline{\boldsymbol{\mu}}_{h,1}, \sqrt{\overline{\boldsymbol{\Sigma}}_{h,11}}\right), \tag{39}$$

and the distribution of $\overline{X}_{h,2}$ is

$$\overline{X}_{h,2} \sim \mathcal{N}\left(\overline{\boldsymbol{\mu}}_{h,2}, \sqrt{\overline{\boldsymbol{\Sigma}}_{h,22}}\right). \tag{40}$$

Furthermore, the integration limits in Eqs. (33) and (34) are implicitly defined as

$$\int_{-\infty}^{b_1(\bar{x}_{h,1})} p_{\overline{X}_1|\overline{X}_2}(u|\bar{x}_{h,1})\mathrm{d}u = \frac{\alpha}{2} \tag{41}$$

$$\int_{b_2(\bar{x}_{h,1})}^{\infty} p_{\overline{X}_1|\overline{X}_2}(u|\bar{x}_{h,1})\mathrm{d}u = \frac{\alpha}{2} \tag{42}$$

$$\int_{-\infty}^{b_1(\bar{x}_{h,2})} p_{\overline{X}_1|\overline{X}_2}(u|\bar{x}_{h,2})\mathrm{d}u = \frac{\alpha}{2} \tag{43}$$

$$\int_{b_2(\bar{x}_{h,2})}^{\infty} p_{\overline{X}_1|\overline{X}_2}(u|\bar{x}_{h,2})\mathrm{d}u = \frac{\alpha}{2}. \tag{44}$$

Finally, the corresponding correlation coefficients in Eqs. (33) and (34) are defined as

$$r_{\overline{X}_{h,2},\overline{X}_{h,1}} = \frac{E\left[(\overline{X}_{h,2} - \overline{\boldsymbol{\mu}}_{h,2})(\overline{X}_{h,1} - \overline{\boldsymbol{\mu}}_{h,1})\right]}{\sqrt{\overline{\boldsymbol{\Sigma}}_{h,22}}\sqrt{\overline{\boldsymbol{\Sigma}}_{h,11}}} \tag{45}$$

$$= r_{\overline{X}_1\overline{X}_2} \frac{\overline{\boldsymbol{\Sigma}}_{11}}{\sqrt{\overline{\boldsymbol{\Sigma}}_{h,11}}\sqrt{\overline{\boldsymbol{\Sigma}}_{h,22}}}, \tag{46}$$

and $r_{\overline{X}_{h,1},\overline{X}_{h,2}} = r_{\overline{X}_{h,2},\overline{X}_{h,1}}$.

The main difference in the evaluation of the $n \geqslant 2$ case compared to the $n = 2$ case stems from the random halving of the sample. Along with the increased number of compromised elements, the halving of the sample randomizes the number of compromised elements in the two halves. As a matter of fact, this kind of random selection is related to the hypergeometric distribution, which describes the probability that in a sample of n distinctive objects j objects are compromised. Therefore, the final error probability β can be defined based on the particular probabilities in Eq. (32) as

$$\beta = \sum_{j=0}^{t} P(t_1 = j)\beta^{(j,t-j)}, \tag{47}$$

where

$$P(t_1 = j) = \frac{\binom{t}{j}\binom{n-t}{\frac{n}{2}-j}}{\binom{n}{\frac{n}{2}}}, \tag{48}$$

is the hypergeometric distribution with parameters n, t, and $\frac{n}{2}$.

To show the gain of our Enhanced Data Aggregation Algorithm compared to a scenario where no attack detection is employed we define d_{imp} as the improvement in the distortion in case of an attack just like in Section 4.1.2 as follows:

$$d_{\mathrm{imp}} = d(Y|A = 1, D = 0) - d(Y|A = 1) \tag{49}$$

$$\cong \frac{1}{n^2}(\tilde{\mu}^2 + \tilde{\sigma}^2) \cdot (1 - \beta), \tag{50}$$

where we still assume that $E|Y_{\mathrm{extr}} - \widehat{Y}|^2$ is close to zero. In Fig. 6, one can see a plot of values of the redefined d_{imp} function for different correlation coefficients represented by the different lines. The subfigures correspond to different attack strengths, i.e., to different number of compromised nodes. The horizontal axes correspond to the expected value $\tilde{\mu}$ of the attacker's distribution, while the vertical axes correspond to the improvement in the distortion d_{imp} defined in Eq. (49). $\tilde{\sigma}$ is considered to be 1, but its value in the range $[0.5, 1.5]$ does not affect the results significantly. The two sequences for even and odd number of compromised nodes are clearly recognizable. In the odd sequence the correlation seems to be a dominating factor, while in the even sequence the law of large numbers improves the attack detection capabilities and thus the value of d_{imp} for less correlated samples.

Secondly, we show how much influence the correlation has on the distortion. In Fig. 7, one can see the distortion $d(Y|A = 1)$ for different values of the correlation coefficient r. The subfigures correspond to different attack strengths, i.e., to different number of compromised nodes. The horizontal axes correspond to the expected value $\tilde{\mu}$ of the attacker's distribution. Here again, assuming that $E|Y_{\mathrm{extr}} - \widehat{Y}|^2$ is close to zero, we can characterize the distortion as

$$\mathrm{d}(Y|A = 1) \cong \frac{1}{n^2}(\tilde{\mu}^2 + \tilde{\sigma}^2) \cdot \beta. \tag{51}$$

The calculations presented in Figs. 6 and 7 are performed with $n = 10$ (i.e., with a 10-nodes network or with a 10-nodes cluster). This small value of n helps in giving an overview of the most probable cases considering the number of nodes an attacker is able to compromise. Moreover, for a smaller sample, the effect of correlation is easier to trace because the compression in the first step of the sample halving approach does not influence the distortion as much as for larger samples. However, we note that the sample halving approach is not restricted in the value of n.

The message of Figs. 6 and 7 is manifold. Firstly, the figures clearly show the effect of the compression step (i.e., halving and aggregating the halves). The random halving of the sample results in different behaviour of the distortion in case the attacker compromises even or odd number

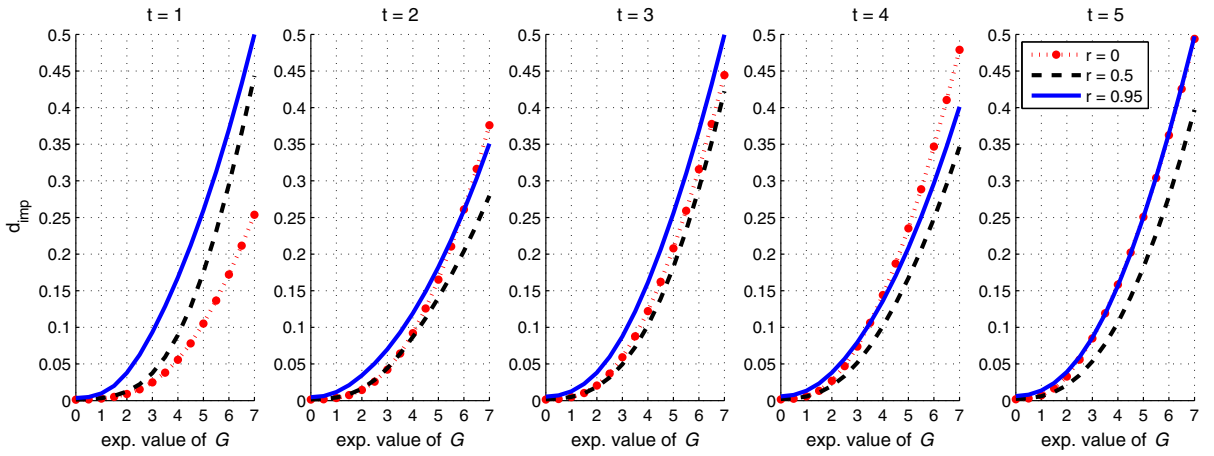


Fig. 6. The improvement in the distortion considering the sample halving approach with $n = 10$ nodes and with different values of r.

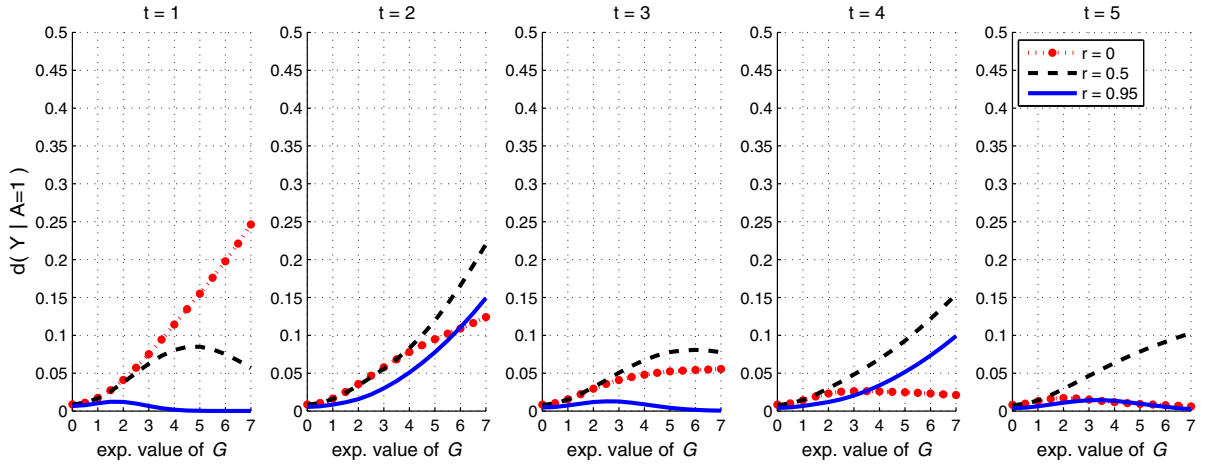


Fig. 7. Distortion caused by the adversary for different values of the correlation coefficient r with $n = 10$ nodes and with $\tilde{\sigma} = 1$.

of elements. The subfigures corresponding to $t = 1$, $t = 3$ and $t = 5$ can be considered as one sequence, while the remaining ones as another sequence. In both figures, the odd sequence consists in three nearly coinciding subfigures on which only the dotted line changes. This indicates that having smaller correlation does not always mean weak resilience in aggregation. However, not considering correlation cannot outperform the correlated case if the correlation coefficient is high enough. The even sequence in the same figures emphasizes the effect of the law of large numbers. Namely, having an uncorrelated (and thus in the Gaussian case independent) sample can be a better base for attack detection than a correlated sample. The explanation for this is that an independent sample is able to narrow very quickly by the means of its standard deviation because of the averaging, while a correlated sample has always a bigger standard deviation. Therefore, the confidence interval calculated based on an independent sample can be very small which then facilitates the detection of outlier elements. The different nature of the odd and even sequences has combinatorial roots. Having even number of compromised elements frequently results in such a halving where exactly the half of the compromised elements are in the first half and the others in the second half. In this case, however, the attack detection capabilities are weak, as the numerical difference between the two sample halves is small. This then introduces a higher distortion and thus a smaller value for d_{imp}. In case of odd compromised elements the halving is always "unfair", one of the halves always possesses more compromised elements than the other, and therefore there is always a remarkable difference between the halves, which implies better attack detection capabilities.

4.2.2. Evaluation with non-constant correlation coefficient

Until now, we 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. Several models have been proposed so far for the calculation of the value

of the correlation coefficient based on these parameters, e.g., the Spherical, the Power Exponential, the Rational Quadratic and the Matérn correlation models [3]. The most widely used correlation model in the literature on spatial statistics is the Power Exponential model [13,27] with several applications [21,25,24,1,19,5], therefore we applied it as well.

Assuming non-constant correlation coefficients the covariance matrix in Eq. (28) will take the following form

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma^2 & r_{12}\sigma^2 & \dots & r_{1n}\sigma^2 \\ r_{21}\sigma^2 & \sigma^2 & \dots & r_{2n}\sigma^2 \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1}\sigma^2 & r_{n2}\sigma^2 & \dots & \sigma^2 \end{pmatrix}, \tag{52}$$

where $r_{ij} = r_{ji}$. r_{ij} can be calculated using the Power Exponential correlation model as

$$r_{ij}(d_{ij}) = \exp\left(-\left(\frac{d_{ij}}{\theta_1}\right)^{\theta_2}\right), \tag{53}$$

where d_{ij} is the Euclidean distance between node$_i$ and node$_j$, θ_1 controls the relation between d_{ij} and r_{ij} with usual values of different integer powers of 10 (i.e., 10, 10^2, ...,10^6), and it depends on θ_2 whether the model is exponential ($\theta_2 = 1$) or squared exponential ($\theta_2 = 2$). For the analysis we have chosen $\theta_1 = 10$ and $\theta_2 = 1$ as in [24,1].

To evaluate the distortion caused by an attacker in the output of the Enhanced Data Aggregation Algorithm in the case of non-constant correlation, one can formulate the probability density function of the correlation coefficient r_{ij} considering uniformly randomly placed sensor nodes as

$$p_{r_{ij}}(x) = \begin{cases} \frac{2\pi\theta_1^2}{\theta_2 x}(-\ln(x))^{\frac{2}{\theta_2}-1} - \frac{8\theta_1^3}{\theta_2 x}(-\ln(x))^{\frac{3}{\theta_2}-1} \\ +\frac{2\theta_1^4}{\theta_2 x}(-\ln(x))^{\frac{4}{\theta_2}-1} \qquad \text{if } x \in \left(\exp\left(-\left(\frac{1}{\theta_1}\right)^{\theta_2}\right), 1\right), \\ \frac{4\theta_1^2}{\theta_2 x}(-\ln(x))^{\frac{2}{\theta_2}-1}\left[\arcsin\left(\frac{2-\theta_1^2(-\ln(x))^{\frac{2}{\theta_2}}}{\theta_1^2(-\ln(x))^{\frac{2}{\theta_2}}}\right)\right. \\ \left.+2\sqrt{\theta_1^2(-\ln(x))^{\frac{2}{\theta_2}}-1}-1\right] - \frac{2\theta_1^4}{\theta_2 x}(-\ln(x))^{\frac{4}{\theta_2}-1} \\ \text{if } x \in \left(Z\exp\left(-\left(\frac{\sqrt{2}}{\theta_1}\right)^{\theta_2}\right), \exp\left(-\left(\frac{1}{\theta_1}\right)^{\theta_2}\right)\right), \\ 0 \text{ if } x \notin \left(\exp\left(-\left(\frac{\sqrt{2}}{\theta_1}\right)^{\theta_2}\right), 1\right). \end{cases} \tag{54}$$

Taking a sample from this distribution (using uniformly random sampling) and applying it to Eq. (52) gives a realistic covariance matrix for a realization of the random node deployment. Then, calculating the conditional p.d.f. $p_{\overline{X}_1|\overline{X}_2}(\cdot|\cdot)$ using this updated covariance matrix and performing the analysis presented in Section 4.2.1 gives the distortion in this given realization.

The conditional p.d.f. $p_{\overline{X}_1|\overline{X}_2}(\cdot|\cdot)$ can be easily described with the help of $\overline{\boldsymbol{\mu}} = \mathbf{M}\boldsymbol{\mu}$ and $\overline{\mathbf{\Sigma}} = \mathbf{M}\mathbf{\Sigma}\mathbf{M}^{\mathrm{T}}$ (see [11]). The correlation coefficient applied in $p_{\overline{X}_1|\overline{X}_2}(\cdot|\cdot)$ can be defined as

$$r_{\overline{X}_1,\overline{X}_2} = \frac{\overline{\Sigma}_{12}}{\sqrt{\overline{\Sigma}_{11}}\sqrt{\overline{\Sigma}_{22}}} \tag{55}$$

The joint probability density functions $p_{\overline{X}_{h,1},\overline{X}_{h,2}}(\cdot,\cdot)$ and $p_{\overline{X}_{h,2},\overline{X}_{h,1}}(\cdot,\cdot)$ can be defined in the same way as in Section 4.2.1.

Repeating the above calculations along with the sampling of $p_{r_{ij}}$ multiple times gives the same result as having multiple sensor networks with different uniformly random deployment. Calculating the average distortion of the repetitions can help us in exposing the characteristic features of this scenario when the correlation coefficient is not constant.

The results of this analysis are very interesting. After performing the repeated sampling and distortion calculation for $t = 1, \dots, 5$ (20 times for each value), the resulting curves are nearly the same as the curves on Fig. 6 and 7 when $r = 0.95$. As it would be difficult to distinguish the two kind of curves in a figure, we show a comparison table consisting of numerical values for the two curves for $t = 2$ (see Table 1). The $t = 2$ choice is confirmed by the fact that the differences are the largest in that case.

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 our previous results: even though we used a simplified scheme in which we 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), our results are still highly relevant when we consider the more realistic scenario of distance-dependent correlation coefficients among the sample elements. Moreover, as the curves for $r = 0$ and $r = 0.95$ are significantly different (for both the d_{imp} and the $\mathrm{d}(Y|A = 1)$ metrics), the latter results also indicate that assuming correlation is a must in order to establish a realistic sensor network data model.

4.2.3. Evaluation assuming a sophisticated attacker

The attacker we considered until now was a simplified one: he added offsets to some of the sensor readings, where the offsets were independent and identically distributed random variables. For the performance evaluation, we categorized the offsets as elements coming from a normal distribution the parameters (i.e., the expected value and the standard deviation) of which are under the control of the attacker. In this section, we investigate the case of a more sophisticated attacker. Namely, we assume that the attacker knows the Enhanced Data Aggregation Algorithm in detail, including the Attack Detection Algorithm $Det(\cdot,\cdot)$.

Table 1
Numerical values of the $r = 0.95$ curve (in Fig. 6) compared with 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

Moreover, the attacker also knows the size of the sample that the base station gathers in a given query, and he can arbitrarily modify the observed sample elements.

Therefore, 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 we 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. We 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.

After calculating the individual distortions for all cases of the halving and all combinations of the compromised measurements, the attacker selects those measurements to compromise, the modification of which leads to the highest distortion on average. As the attacker cannot influence 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.

We simulate the sophisticated attacker assuming that the original sample is normally distributed (with parameters $\mu = 0$, $\sigma = 1$ and $n = 10$) and correlated. The correlation of the sample is modelled with the Power Exponential correlation model with parameters $\theta_1 = 10$ and $\theta_2 = 1$ as in [24,1]. We perform simulations for two different attacker behaviours. The first behaviour is when the attacker perturbs some of the sample elements with an offset, while the second behaviour describes the case when the attacker replaces some of the sample elements with a common maximum. As the resulting figures for the two behaviours are quite similar, we only detail the results considering the first behaviour in Fig. 8.

Fig. 8 shows a simulation result (i.e., not an analytical calculation like all the figures until now). The horizontal axes correspond to the offset value chosen by the attacker, while the vertical axes correspond to the distortion in the aggregate. The five subfigures correspond to different number of observed sensor measurements. As the original samples are drawn randomly for the simulations, the curves in the subfigures are somewhat irregular.

In the first three subfigures (i.e., up to 30% compromised nodes), the highly correlated measurements imply smaller distortion than the independent measurements (similarly to the $t = 1$ and $t = 2$ subfigures in Fig. 7). 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 (similarly to the $t = 4$ subfigure in Fig. 7). 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 $\alpha = 0.1$.

As one can see, the subfigures corresponding to $t = 2$ and $t = 3$ show similarities, and the same happens in the case of subfigures corresponding to $t = 4$ and $t = 5$. In general, the attacker cannot reach a significantly higher distortion by compromising $2k + 1$ sensor readings compared to the case when compromising only $2k$ sensor readings (maybe except for $k = 1,\ r = 0$). The reason for this property is, on the one hand, that the attacker is able to choose the number of measurements he is going to compromise. For example, it is possible for an attacker to observe three sample elements but compromise only two of them. On the other hand, the random halving step in $Det(\cdot,\cdot)$ has a high influence on the result, as even numbered compromised

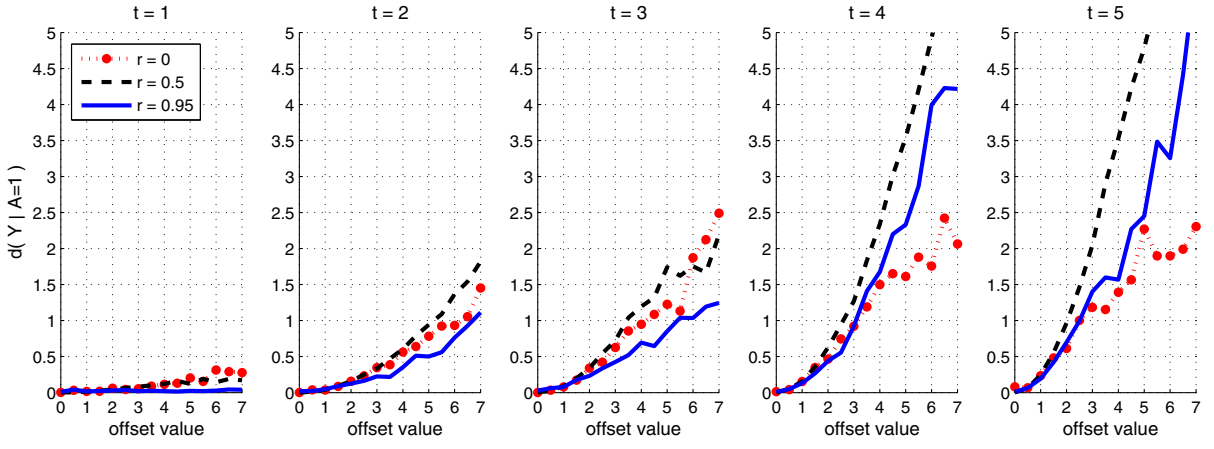


Fig. 8. Distortion caused by a sophisticated adversary for different values of the correlation coefficient r.

elements can be halved in a way that both halves contain the same number of compromised elements, which weakens the attack detection capabilities, while odd numbered elements cannot be halved in such a "fair" way, which result in better attack detection capabilities.

The results for the distortion caused by a sophisticated attacker can be summarized as these are highly related to the analytical results in Section 4.2.1 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. Nevertheless, we note that the sophisticated attacker is still not an optimal attacker, and thus, the results presented in this section do not correspond to the worst case.

After having presented the results of our sample halving approach, having illustrated the impact of correlation on resilient aggregation, having verified our results considering realistic correlation coefficient distribution scenarios and a sophisticated attacker as well, in the next section, we present some possible extensions to the work presented.

5. Discussion

5.1. Does this approach allow in-network processing?

The concept of performing the aggregation at the base station allows us to get rid of some typical "networking" problems (like e.g., routing, lost messages, etc.) and to concentrate on the novel statistical framework presented. However, our scheme can support in-network aggregation as well. There are two straightforward ways to perform in-network aggregation in our case: Firstly, aggregator nodes chosen among the sensor nodes can aggregate the measurements of the sensors in their clusters. Algorithms 1 and 2 are both very energy-efficient as they do not require additional communication, thus, they can run even on resource-constrained sensor nodes. After the aggregation, the aggregator nodes send the result to the base station, and the base station can average them without further investigation, as the analysis has been already done by the aggregator nodes. The drawback of this way of processing is that the aggregator nodes have to decrypt the messages of their corresponding clusters as our algorithms need raw data as their input.

Secondly, considering again that the algorithms run on the base station and that the Attack Detection Algorithm only needs two averages in order to make its decision, the aggregator nodes only have to sort the measurements into two groups randomly, sum up these groups, and send only the sums to the base station. Upon reception of the sums the base station is able to calculate the averages by, again, sorting the received sums into two groups randomly, summing them up, and dividing the two sums by the total number of measurements they are based upon. Having the final averages, the base station is now able to perform the Enhanced Data Aggregation Algorithm. This latter approach has the advantage that the messages do not have to be decrypted by the aggregator nodes while they perform the summation. The tool that allows to sum up encrypted data is called 'homomorphic encryption' (see [12]). Moreover, both approaches fulfill the requirement of having the minimum number of messages transmitted (wireless transmission consumes a plenty of energy), as aggregation invokes compression of the data too.

5.2. How to relax the knowledge about the conditional p.d.f.?

In the previous sections, we assumed that the conditional p.d.f. $p_{X_1|X_2}(\cdot|\cdot)$ (or $p_{\overline{X}_1|\overline{X}_2}(\cdot|\cdot)$) is known to the Attack Detection Algorithm. In the following, we will show how our algorithm behaves in case this conditional p.d.f. is not precisely known. Let us assume that the Attack Detection Algorithm knows only $\hat{p}_{X_1|X_2}(x|y) = p_{X_1|X_2}(x|y) + \Delta(x|y)$, where $\int_{-\infty}^{\infty} |\Delta(x|y)|\mathrm{d}x < \delta$ 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, $\int_{-\infty}^{\infty} \Delta(x|y)\mathrm{d}x = 0$ for any y. The imprecise knowledge implies a wider confidence interval in Algorithm 1 with upper and lower bounds $\hat{b}_1(\cdot)$ and $\hat{b}_2(\cdot)$ (see Eqs. (6)–(9)).

As $\int_{-\infty}^{\infty} \Delta(x|y)\mathrm{d}x = 0$, 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 Eqs. (6)–(9), this would imply

$$\int_{-\infty}^{\hat{b}_1(z)} p_{X_1|X_2}(u|z)\mathrm{d}u = \frac{\alpha}{2} - \frac{\delta}{4} \tag{56}$$

$$\int_{\hat{b}_2(z)}^{\infty} p_{X_1|X_2}(u|z)\mathrm{d}u = \frac{\alpha}{2} - \frac{\delta}{4} \tag{57}$$

and two similar equations with x_2 instead of z. 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. on the distortion just like in Section 4.1.2. (We note, however, that Eqs. (56) and (57) implicitly upper bound δ by 2α.)

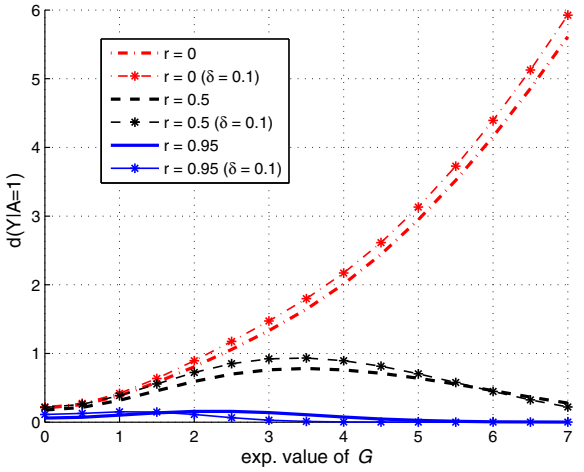


Fig. 9. The effects of the imprecise knowledge of the p.d.f. on the distortion.

Fig. 9 shows the results of this evaluation for $\delta = 0.1$ and $n = 2$. As expected, the imprecise knowledge of the conditional p.d.f. 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. Especially, for $r = 0.95$ the attack detection capabilities become better for $\tilde{\mu} \geqslant 1.5$, which emphasizes again the important role of correlation. Generally speaking, the lack of precise information about the conditional p.d.f. does not alter our previous results significantly when assuming a moderate δ, while it can also be beneficial for higher correlation strength.

5.3. What is the optimal attack against the proposed scheme?

An optimal attacker is defined as an attacker who can reach the highest possible distortion at the output of the aggregation function. We already presented two kind of attackers: a simplified one in Section 4.1.2 in order to carry out the analysis, and a sophisticated one in Section 4.2.3 in order to demonstrate the capabilities of the proposed scheme in a more general setting. However, none of these attackers are optimal attackers, as both contain some restrictions considering the way how they perform the attack. More work needs to be done for identifying the optimal attack against the algorithms presented in this paper, and for evaluating the performance of those algorithms when they face the optimal attacker.

6. Conclusion and future work

In this paper, we were concerned with a serious threat against sensor networks that consists in altering the measured parameters of the environment around the sensor nodes. We proposed a resilient data aggregation framework, called CORA, that mitigates this problem. The novelty of CORA is that it takes advantage of the naturally existing correlation between the sensor readings reported to the base station; in particular, correlation is exploited to increase the probability of attack detection, which in turn, is used to decrease the distortion caused by an attack at the output of the aggregation function. We emphasize that the operation of CORA does not depend on any particular assumptions on the distribution of the sensor readings nor on the distribution of the measurement offset introduced by the attacker. We evaluated the effectiveness of CORA both formally and by means of simulation by characterizing its false positive and false negative probabilities along with the final distortion in the aggregate. The results show that CORA can significantly decrease the distortion and that the level of improvement offered by CORA increases as the correlation increases considering typical attacks (i.e., when the number of compromised measurements is low).

During the development of our resilient data aggregation scheme, we identified some interesting future research directions. One of them is to consider the case when the conditional p.d.f. used for attack detection can become outdated (e.g., the temperature changes when heading from winter to spring, therefore different temperature values has to be labelled as outlier than before). The effect of this could probably be modelled with a similar approach to what was presented in Section 5, but with a time-dependent uncertainty. Searching for the optimal attacker is very important as well, as already mentioned in the previous section. Furthermore, we intend to work on improving the framework presented to be applicable in sample filtering, i.e., instead of dropping the compromised sample one could filter out the compromised elements. We believe that with sample filtering we will be able to further reduce the distortion of the aggregate considering any kind of attacks.

Acknowledgements

The work described in this paper is based on results of the IST FP6 STREP UbiSec&Sens (www.ist-ubisecsens.org). UbiSec&Sens receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. The second author has been partially supported by the HSN Lab.

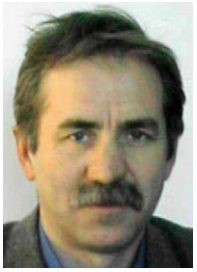
References

[1] I.F. Akyildiz, M.C. Vuran, O.B. Akan, On exploiting spatial and temporal correlation in sensors networks, in: Proceedings of the Second Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2004.
[2] T. Arampatzis, J. Lygeros, S. Manesis, A survey of applications of wireless sensors and wireless sensor networks, in: Proceedings of the 13th Mediterranean Conference on Control and Automation, 2005.
[3] J.O. Berger, V.D. Oliveira, B. Sansó, Objective Bayesian analysis of spatially correlated data, Journal of the American Statistical Association 96 (456) (2001) 1361–1374.
[4] BriMon: Wireless Sensor Network based Bridge Monitoring, <http://www.cse.iitk.ac.in/users/braman/brimon.html>.
[5] G. Bravos, A.G. Kanatas, A. Kalis, Lifetime evaluation and spatial correlation effects on wireless sensor networks, in: Proceedings of 15th IST Mobile & Wireless Communications Summit, 2006.
[6] T. Brooke, J. Burrell, From ethnography to design in a vineyard, in: Proceedings of the Conference on Designing for User Experiences, 2003.
[7] J. Burrell, T. Brooke, R. Beckwith, Vineyard computing: sensor networks in agricultural production, IEEE Pervasive Computing 3 (1) (2004) 38–45.
[8] L. Buttyán, P. Schaffer, I. Vajda, RANBAR: RANSAC-based resilient aggregation in sensor networks, in: Proceedings of the Fourth ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN), 2006.
[9] L. Buttyán, P. Schaffer, I. Vajda, Resilient aggregation with attack detection in sensor networks, in: Proceedings of the Second IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS), 2006.
[10] V. Chatzigiannakis, S. Papavassiliou, Diagnosing anomalies and identifying faulty nodes in sensor networks, IEEE Sensors Journal 7 (5) (2007) 637–645.
[11] M.H. DeGroot, Optimal Statistical Decisions, John Wiley & Sons, 2004.
[12] J. Domingo-Ferrer, A provably secure additive and multiplicative privacy homomorphism, in: Proceedings of the Fifth International Conference on Information Security (ISC), 2002.
[13] T. Gneiting, M. Genton, P. Guttorp, Geostatistical Space-Time Models, Stationarity, Separability, and Full Symmetry, Chapman & Hall/CRC, Boca Raton, FL, USA, 2007.

[14] A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, in: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), 2004.
[15] R.-G. Lee, K.-C. Chen, C.-C. Lai, S.-S. Chiang, H.-S. Liu, M.-S. Wei, A backup routing with wireless sensor network for bridge monitoring system, Elsevier Measurement 40 (2007) 55–63.
[16] T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos, Distributed deviation detection in sensor networks, ACM SIGMOD Record 32 (4) (2003).
[17] S. Papadimitiou, H. Kitagawa, P.B. Gibbons, LOCI: fast outlier detection using the local correlation integral, in: Proceedings of the 19th International Conference on Data Engineering (ICDE), 2003.
[18] K. Piotrowski, P. Langendoerfer, S. Peter, How public key cryptography influences wireless sensor node lifetime, in: Proceedings of the Fourth ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN), 2006.
[19] T. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, Upper Saddle River, NJ, USA, 2001.
[20] 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.
[21] G.L. Stüber, Principles of mobile communication, 2nd ed., Kluwer Academic Publishers., Norwell, MA, USA, 2001.
[22] S. Tanachaiwiwat, A. Helmy, Correlation analysis for alleviating effects of inserted data in wireless sensor networks, in: Proceedings of Mobile and Ubiquitous Systems: Networking and Services, 2005.
[23] Pickberry Vineyard, <https://www.accenture.com/global/services/accenture_technology_labs/r_and_i/pickberry.htm>.
[24] M.C. Vuran, O.B. Akan, I.F. Akyildiz, Spatio-temporal correlation: theory and applications for wireless sensor networks, Elsevier Computer Networks 45 (3) (2004) 245–259.
[25] M.C. Vuran, I.F. Akyildiz, Spatial correlation-based collaborative medium access control in wireless sensor networks, IEEE/ACM Transactions on Networking (TON) 14 (2) (2006) 316–329.
[26] D. Wagner, Resilient aggregation in sensor networks, in: Proceedings of the Second ACM Workshop on Security of Ad hoc and Sensor Networks (SASN), 2004.
[27] R.O. Weber, P. Talkner, Some remarks on spatial correlation function models, Monthly Weather Review 121 (9) (1993) 2611–2617.
[28] F. Ye, H. Luo, S. Lu, L. Zhang, Statistical en-route filtering of injected false data in sensor networks, IEEE Journal on Selected Areas in Communications 23 (4) (2005) 839–850.
[29] S. Yoon, C. Shahabi, Exploiting spatial correlation towards an energy efficient clustered aggregation technique (CAG), in: Proceedings of the IEEE International Conference on Communications (ICC), 2005.
[30] Y. Zhu, R. Vedantham, S. Park, R. Sivakumar, A scalable correlation aware aggregation strategy for wireless sensor networks, in: Proceedings of the First International Conference on Wireless Internet (WICON), 2005.

Levente Buttyán received the M.Sc. degree in Computer Science from the Budapest University of Technology and Economics (BME) in 1995, and the Ph.D. degree from the Swiss Federal Institute of Technology, Lausanne (EPFL) in 2002. In 2003, he joined the Department of Telecommunications at BME, where he currently holds a position as Associate Professor and works in the Laboratory of Cryptography and Systems Security (CrySyS). His research interests are in the design and analysis of security protocols for wired and wireless networks, including wireless sensor networks and ad hoc networks. More information is available at http://www.hit.bme.hu/~buttyan/.

Péter Schaffer received the M.Sc. degree in Computer Science from the Budapest University of Technology and Economics (BME) in 2005. During his M.Sc. he joined the Laboratory of Cryptography and Systems Security (CrySyS) in 2004. Since 2005 he is a Ph.D. student at the same laboratory under the supervision of Levente Buttyán. His research interests are in security of wireless sensor networks, mainly focused on resilience and statistical approaches. More information is available at http://www.crysys.hu/members/pschaffer/.



István Vajda is a Professor at the Department of Telecommunications, Budapest University of Technology and Economics (BME). He is the head of the Laboratory of Cryptography and Systems Security (CrySyS). His research interests are in Cryptography and Coding Theory. He has teaching experience in Algebraic Coding Theory, Cryptography, and Information Theory. More information is available at http://www.crysys.hu/members/ivajda/.