Resilient Aggregation with Attack Detection 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, Hungary
{buttyan, schaffer, vajda}@crysys.hu

Abstract

In this paper, we propose a new model of resilient data aggregation in sensor networks, where the aggregator analyzes the received sensor readings and tries to detect unexpected deviations before the aggregation function is called. In this model, the adversary does not only want to cause maximal distortion in the output of the aggregation function, but it also wants to remain undetected. The advantage of this approach is that in order to remain undetected, the adversary cannot distort the output arbitrarily, but rather the distortion is usually upper bounded, even for aggregation functions that were considered to be insecure earlier (e.g., the average). We illustrate this through an example in this paper.

1. Introduction

The problem of resilient data aggregation is to perform data aggregation in the presence of an adversary that can modify the input to the aggregation function. In fact, there are two ways in which the input can be modified. Firstly, the messages that carry the data from the sensors to the place of aggregation (usually the base station) can be modified in transit. This can be detected by cryptographic techniques, and resilient data aggregation is not concerned with this problem. Secondly, the adversary may compromise some sensors in the network and affect their readings (e.g., it can increase the temperature around a temperature sensor). This latter kind of attack cannot be prevented, neither detected, by cryptographic mechanisms. Resilient aggregation is concerned with this problem.

The term resilient aggregation has been coined by David Wagner in his SASN 2004 paper [5]. In that paper, Wagner investigates the following question: Which aggregation functions can be securely and meaningfully computed in the presence of a few compromised sensors? The rather bad news of Wagner's paper is that some of the very useful and widely used aggregation functions, such as the average, the minimum, and the maximum, are inherently insecure, which means that an adversary can cause arbitrary distortion in the aggregated value by modifying only a small number of sensor readings. Wagner proposes to use the median instead of the average, which is a more robust aggregation function. He also proposes trimming as a mechanism that can help to achieve resilience.

Apart from the idea of trimming, Wagner assumes that the data is fed into the aggregation function immediately, without prior analysis. In this paper, we relax this assumption. More precisely, we propose a novel data aggregation model, where the aggregator analyzes the input data before aggregation, and tries to detect unexpected deviations in the received sensor readings. (In fact, trimming is a special case of this more general idea.) In our model, the adversary does not only want to cause maximal distortion in the output of the aggregation function, but it also wants to remain undetected. We show that in this case, the distortion caused by the adversary can usually be upper bounded, even for aggregation functions that were considered to be insecure by Wagner in [5] (e.g., the average). This result has high practical importance, since these functions are commonly used in practice.

We must emphasize that attack detection is possible only if the detector has some *a priori* knowledge about the system. For instance, an unusually high value can be detected only if one knows what the "usual value" should be. We note, however, that assuming some a priori knowledge is reasonable in most of the practical applications. It might be known, for instance, that the distribution of the observed random variable belongs to a particular family of distributions, even if the actual parameters of the distribution are not known. Our general idea is, therefore, to take advantage of this a priori knowledge by letting the detector check if the received data is consistent with it. Clearly, the actual algorithm to be used for checking consistency depends on the nature of the knowledge available; we present a detailed example later in the paper.

The rest of the paper is organized as follows: In Section 2, we introduce our model of data aggregation with attack detection. Then, in Section 3, we study the properties of the model through a specific example. In Section 4, we report on some related works. Finally, in Section 5, we conclude the paper.

2. Model

Our model of data aggregation with attack detection is illustrated in Figure 1. We 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. We 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. We assume that X_i $(i = 1, 2, \ldots, n)$ are identically distributed and independent. $\bar{X} = (X_1, X_2, \ldots, X_n)$ is the vector that contains the readings of all sensors.

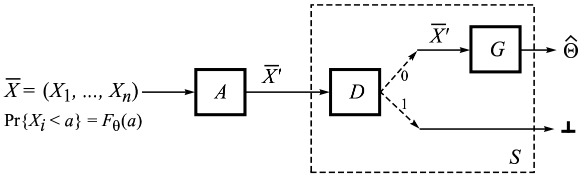
$\bar{X} = (X_1, \ldots, X_n)$
$\Pr\{X_i < a\} = F_\theta(a)$
A
$\bar{X}'$
D
0
1
$\bar{X}'$
G
$\hat{\Theta}$
⊥
S

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 $\bar{X}'$.

The aggregation procedure S has two steps. First, input $\bar{X}'$ is analyzed in order to detect attacks. This is modelled by a function D, which outputs 1 if an attack is detected, and 0 otherwise. We assume that if an attack is detected, then $\bar{X}'$ is thrown away. If no attack is detected, then the processing continues with executing the aggregation function G on input $\bar{X}'$, which outputs the aggregated value $\hat{\Theta}$. Formally, the operation of S is described as follows:

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

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

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

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

In addition, we assume that the adversary does not want to be detected, or more precisely, the adversary wants to keep the probability of successful detection of an attack under a given value p^*:

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

We assume that the adversary knows the detection algorithm D (including the a priori knowledge used in the algorithm) and the aggregation function G. We further characterize the adversary by the number $k < n$ of sensors that it has compromised. This means that $\bar{X}$ and $\bar{X}'$ can differ in at most k positions. Finally, following [5], we may distinguish *omniscient* and *myopic* adversaries. An omniscient adversary first observers the readings of all n sensors, then chooses k sensors to attack and modifies their readings. A myopic adversary can only observe and modify the readings of k sensors that are selected before the attack.

3. Attack detection using sample halving

In this section, we illustrate through an example how attack detection can be implemented, and how useful it can be in upper bounding the distortion achievable by the adversary.

We consider a myopic adversary, which can observe and modify the readings of $k \ll n$ sensors (selected before the attack). The adversary attacks by adding a constant value $m > 0$ to the reading of each selected sensor. Therefore, $X'_i = X_i$ for the non-compromised sensors, and $X'_i = X_i + m$ for the compromised ones. Of course, it is not known which sensors are compromised.

Recall the assumption that the sensor readings X_i $(1 \leq i \leq n)$ are independent and identically distributed. We assume that nothing is known about this distribution except for the fact that its variance is 1. We assume that the goal of the aggregator is to estimate the mean θ, and therefore a natural choice for the aggregation function is the average:

$$G(\bar{X}') = \frac{1}{n}\sum_{i=1}^{n} X_i' \tag{4}$$

Then, the distortion d achieved by the adversary can be computed as follows:

$$d = \frac{k \cdot m}{n} \tag{5}$$

The attack detector uses the following algorithm. It first computes $Z_1 = X_1' + \ldots + X_{n/2}'$ and $Z_2 = X_{n/2+1}' + \ldots + X_n'$, where, for simplicity, we assume that n is even, and then it computes $W = Z_1 - Z_2$. It is known that if there was no attack, then the distribution of W would be $\mathcal{N}(0, \sqrt{n})$, the Gauss distribution with mean 0 and variance $\sqrt{n}$. 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:

$$D(\bar{X}') = \begin{cases} 1 & \text{if } |W| > h_\alpha \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

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):

$$\Pr\{|W| > h_\alpha \mid H_0\} = 2 - 2 \cdot \Phi(h_\alpha/\sqrt{n}) = \alpha \tag{7}$$

The relationship of h_α and α is illustrated in Figure 2.

Now, we will determine the probability of detection in the case when there is an attack (H_1 hypothesis). We use the following simple observation: in this example, the expected value $\mathbb{E}[W]$ of W is a multiple of m, and it lies in the interval $[-km, km]$. Indeed, if k_1 denotes the number of compromised readings in the first half $X_1', \ldots, X_{n/2}'$ of the readings, and k_2 denotes the number of compromised readings in the second half $X_{n/2+1}', \ldots, X_n$ of the readings, where $k_1 + k_2 = k$, then

$$\begin{aligned} \mathbb{E}[W] &= \mathbb{E}[X_1' + \ldots + X_{n/2}'] - \mathbb{E}[X_{n/2+1}' + \ldots + X_n'] \\ &= (\frac{n}{2} \cdot \theta + k_1 \cdot m) - (\frac{n}{2} \cdot \theta + k_2 \cdot m) \\ &= (k_1 - k_2) \cdot m \end{aligned}$$

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

$$\Pr\{D(\bar{X}') = 1 \mid H_1\} = \tag{8}$$
$$\sum_{\ell=-k}^{k} \Pr\{|W| > h_\alpha \mid \mathbb{E}[W] = \ell m\} \cdot \Pr\{\mathbb{E}[W] = \ell m\}$$

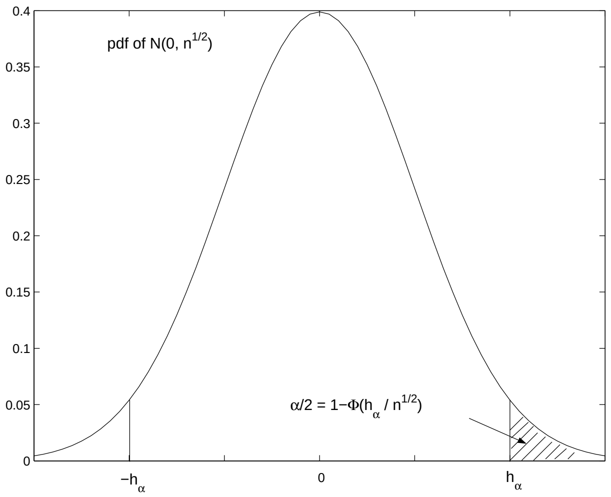

Figure 2. The value of h_α is determined by the probability α of false detection in the case when there is no attack, which corresponds to the tails of the distribution $\mathcal{N}(0, \sqrt{n})$

The first factor of the terms in the above sum can be easily computed using the fact that if $\mathbb{E}[W] = \mu$, then the distribution of W is $\mathcal{N}(\mu, \sqrt{n})$ (see also Figure 3 for illustration):

$$\begin{aligned} &\Pr\{|W| > h_\alpha \mid \mathbb{E}[W] = \mu\} = \\ &= \Pr\{W > h_\alpha \mid \mathbb{E}[W] = \mu\} + \\ &\quad \Pr\{W < -h_\alpha \mid \mathbb{E}[W] = \mu\} \\ &= 1 - \Phi\left(\frac{h_\alpha - \mu}{\sqrt{n}}\right) + \Phi\left(\frac{-h_\alpha - \mu}{\sqrt{n}}\right) \end{aligned} \tag{9}$$

In order to compute the second factor, we assume that the sample is divided into two halves in a random manner (or equivalently, that the adversary compromises sensors in a random manner, and it does not know in advance, in which halves the compromised sensors will fall when the detection algorithm is run). Therefore, the probability of the event $\mathbb{E}[W] = \ell m$ is equal to the probability that the difference between the number of compromised sensors in the two halves is ℓ, when the sample is halved randomly. This can be calculated using basic combinatorics:

- if k is odd, then $\Pr\{\mathbb{E}[W] = \ell m\} = 0$ if ℓ is even, and

$$\Pr\{\mathbb{E}[W] = \ell m\} = \frac{\begin{pmatrix} k \\ \frac{k+\ell}{2} \end{pmatrix} \cdot \begin{pmatrix} n-k \\ \frac{n}{2} - \frac{k+\ell}{2} \end{pmatrix}}{\begin{pmatrix} n \\ \frac{n}{2} \end{pmatrix}} \tag{10}$$

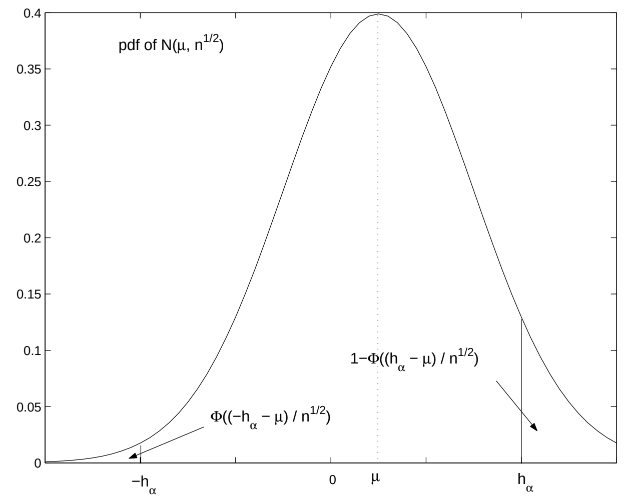

Figure 3. The distribution of W in the case when there is an attack is $\mathcal{N}(\mu, \sqrt{n})$, where $\mu = \mathbb{E}[W]$. The probability of detection corresponds to the tails of this distribution.

if ℓ is odd and $-k \leq \ell \leq k$;

- if k is even, then $\Pr\{\mathbb{E}[W] = \ell m\} = 0$ if ℓ is odd, and

$$\Pr\{\mathbb{E}[W] = \ell m\} = \frac{\binom{k}{\frac{k+\ell}{2}} \cdot \binom{n-k}{\frac{n}{2} - \frac{k+\ell}{2}}}{\binom{n}{\frac{n}{2}}} \tag{11}$$

if ℓ is even and $-k \leq \ell \leq k$.

Using the above formulae and expression (9), we can calculate (8) for given values of the parameters. Figure 4 illustrates the result of this calculation for $n = 100$ and $\alpha \approx 0.05$ (which gives $h_\alpha = 20$). The different curves belong to different values of k.

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.

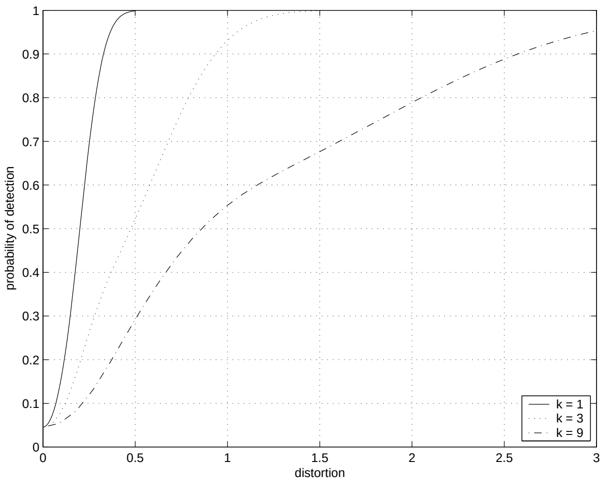

Figure 4. The attack detection probability as a function of the distortion achieved by the adversary for $n = 100$ and $\alpha \approx 0.05$ (which gives $h_\alpha = 20$). The different curves belong to different values of k.

3.1. Generalization

When trying to distort the distribution parameters (mean, median, etc.), the attacker puts more weight into one of the tails of the probability density function, which leads to an asymmetric, skewed distribution. In some sense, the sample halving technique described above tries to detect this asymmetry. The idea of detecting asymmetry can be generalized as follows. It is well known that the standard characterizing parameters of a distribution are the mean (or median), the standard deviation, the kurtosis, and the skewness. Here, we emphasize the role of the skewness, which measures the asymmetry of the distribution. It is reasonable to assume that some empirical distribution of the skewness in case there is no attack is available, since normally, the system is not attacked, most of the time. The attack detection algorithm can compare the skewness of the sample to the expected value of the skewness in order to decide if an attack is taking place or not.

4. Related work

Resilient aggregation has strong connections to robust statistics as this has been pointed out in [5]. Robust statistical techniques have been used in sensor networks for various tasks, such as localization [3] and calibration [1], but they have not been applied in the context of data aggregation in sensor networks.

Wagner's paper [5] considers the problem of resilient data aggregation in sensor networks. However, that paper has a different model, where the base station "blindly" starts to compute the aggregation function on the received data without trying to detect if an attack has taken place. We attempt to detect attacks before starting the aggregation, which allows us to use a wider range of aggregation functions.

Finally, another set of related works [2, 4] is concerned with the application of cryptographic techniques, such as encryption and authentication, in order to prevent that forged data are input into the aggregation function. While those techniques allow the detection of an adversary that modifies the data packets that carry the sensor readings, they cannot be used to detect the type of adversaries considered in this paper that can falsify the sensor readings before they are placed in the data packets.

5. Conclusion and future work

In this paper, we proposed a new model of resilient data aggregation in sensor networks, where the aggregator analyzes the received sensor readings and tries to detect unexpected deviations before the aggregation function is called. We assumed that the adversary does not only want to distort the output of the aggregation function, but it also wants to remain undetected. We showed that under this assumption, the achievable distortion is usually upper bounded, even for aggregation functions that were considered to be insecure earlier (e.g., the average).

We illustrated our approach through a specific example where the attack detection algorithm splits the received sample into two halves, and these halves are checked against each other. We have also performed the calculations for the case when the attack detection algorithm uses the χ^2-test to check the received sample against a hypothetical distribution, and one parameter of the distribution is estimated from the sample itself. However, due to space limitations, this latter calculation has not been included in this paper.

The work presented in this paper is a work-in-progress. We intend to further study the behavior of our model through other examples. Another interesting future direction that we intend to explore is to consider redundant, or highly correlated sensor readings. We believe that assuming correlated measurements will further limit the capabilities of the adversary, as attack detection becomes easier, especially if the adversary is not aware of which sensors are correlated.

6. Acknowledgement

The work presented in this paper has partially been supported by the the UbiSec&Sens Project (IST-026820) and the Hungarian Scientific Research Fund (T046664). The first author has been further supported by IKMA and by the Hungarian Ministry of Education (BÖ2003/70). The second author has been further supported by the High Speed Networks Laboratory (HSN Lab).

References

[1] V. Bychkovskiy, S. Megerian, D. Estrin, M. Potkonjak. A collaborative approach to in-place sensor calibration. In *Proceedings of the Second International Workshop on Information Processing in Sensor Networks (IPSN)*, 2003.

[2] L. Hu, D. Evans. Secure aggregation for wireless networks. In *Proceedings of the Workshop on Security and Assurance in Ad hoc Networks*, January 2003.

[3] Z. Li, W. Trappe, Y. Zhang, B. Nath. Robust statistical methods for securing wireless localization in sensor networks. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*, April 2005.

[4] B. Przydatek, D. Song, A. Perrig. SIA: Secure Information Aggregation in Sensor Networks. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2003.

[5] D. Wagner. Resilient aggregation in sensor networks. In *Proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN)*, October 2004.