# Assuring Trustworthiness of Sensor Data for Cyber-Physical Systems

Björn Stelte and Gabi Dreo Rodosek
Faculty of Computer Science
Universität der Bundeswehr München
85577 Neubiberg, Germany
Email: {bjoern.stelte, gabi.dreo}@unibw.de

***Abstract*—The term cyber-physical system (CPS) refers to the tight conjoining of and coordination between computational and physical resources. A typical Wireless Sensor Network (WSN) that consists of hundreds off-the-shelf cheap sensor nodes is a common application of a CPS. Each sensor node is equipped with a power-efficient micro-controller, a wireless transmitter, and sensory for environmental monitoring. WSNs are used for monitoring critical infrastructures or habitat monitoring as well as used in military scenarios for urgent decision making. A precondition for such a decision support is to assure the trustworthiness of the reported sensor data. Simply securing the hardware is difficult due to existing resource limitations; in particular power consumption and lack of tamper resistance. A WSN, however, consists of several hundreds of sensor nodes. The idea is to use this redundancy, which is an inherent feature of WSN, to assure trustworthiness. Until now, device redundancy has been used for assuring fault tolerance only but not for security purposes. In this paper, we propose to use device redundancy in WSN to detect and isolate malicious nodes, and with this efficiently protect off-the-shelf WSN as well as assure trustworthiness of sensor data.**

## I. Introduction

A Cyber-Physical System (CPS) integrates computation, networking and physical processes. Typically, they are designed as networks where sensors, actuators and embedded devices are interconnected to sense, monitor, and control the physical world. It is argued that CPS applications have the potential to eclipse the 20th century IT innovations. CPS addresses multiple sectors of industry such as automotive, aviation, healthcare, public transportation, critical infrastructure, and use mostly information collected by Wireless Sensor Networks (WSN).

A WSN consists mostly of hundreds of cheap and resource-limited sensor devices [1]. Trustworthiness of the collected and transmitted data as well as the robustness of the WSN is hereby essential. In addition, the specifics of WSN such as the enormous amount of sensor devices on one side and resource constraints (battery, processing power etc.) on the other side, need to be taken into account when developing concepts to assure trustworthiness of sensor data. Traditional approaches such as hardening the hardware have limitations in their applicability. The enormous amount of sensor devices provides, however, an additional benefit. We propose to use the inherent redundancy of WSN to identify malicious nodes and to assure the trustworthiness of the collected and transmitted data. Furthermore, we propose to cluster sensor nodes, and propose *virtual cluster head* nodes as part of our concept. A WSN where sensors are clustered and assigned to virtual cluster heads, and where each cluster communicates with the multi-cluster head (gateway) is depicted in Figure 1. Each cluster is treated as a virtual sensor node for the upper layer gateway, thus the gateway node acts also as a multi-cluster head. We have such a topology in mind when discussing the problem area.

The paper is structured as follows: Section 2 gives an introduction to the problem area. An analysis of the related work and the identification of the gaps are presented in Section 3. Section 4 is introduces the proposed approach for assuring trustworthiness by using the Dempster-Shafer (DS) evidence theory. An assessment of the approach is presented in Section 5 and a proof-of-concept in Section 6 provides a validation of the proposed concept. Finally, Section 7 concludes the paper.

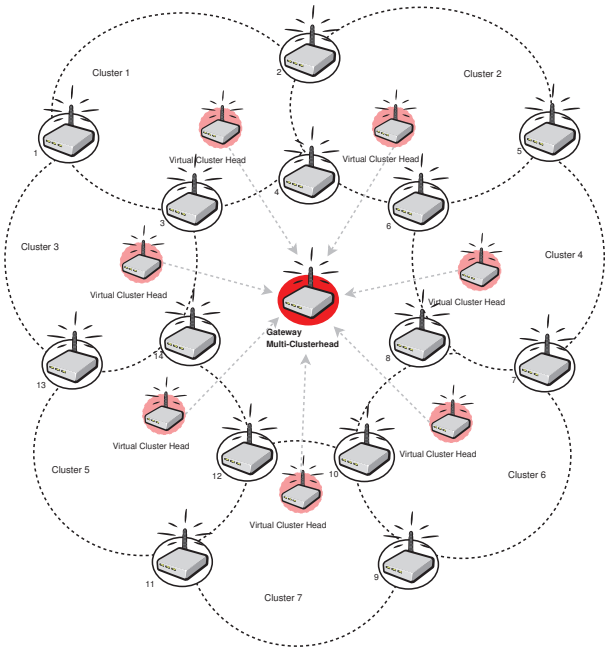



Fig. 1. Sensors are clustered and assigned to virtual cluster heads

## II. Problem Description

A CPS is typically designed as a network, in which sensors, actuators, and embedded devices are networked to sense, monitor, and control the physical world. Thus, CPS are

978-3-901882-50-0 

integrations of computation and physical processes, and are used in various domains (e.g., automotive, aviation, healthcare, public transportation, critical infrastructures). An essential part of a CSP is the data, collected mostly by WSN where each device is equipped with a sensor to monitor physical or environmental conditions. A WSN consists of self-organized sensor nodes, which are small devices that react on changes in the monitored environment. They are used to cooperatively monitor large environments, such as critical infrastructures in gas and oil industries or even for battlefield surveillance in military scenarios. In such scenarios, trustworthy data aggregation is essential as a basis to build a trustworthy decision making CPS application.

Critical infrastructures such as Smart Grids are lucrative targets for cyber attacks. Thus, it is not surprisingly that cyber attacks on CPS-based environments are not a fiction anymore [2]. Key security challenges that apply in order to protect networks against cyber attacks (e.g., eavesdropping, injection, modification of packets, node capturing) and assure privacy, data integrity, data freshness as well as availability need to be addressed for CPS-based environments, resp. WSN, as well.

CPS and WSN have, however, additional requirements that need to be taken into account when developing security concepts. An example for this is the fact that patching and frequent updates are not well-suited for control systems [2].The real-time requirement is another characteristic. However, the real-time ability of a control system is an attacker vector, too. Packet delay or retransmission attacks are used to disturb control systems. Furthermore, the real-time availability is harder to guarantee than service availability in traditional systems. CPS and WSN sensor nodes are limited in their energy, computation, and communication capabilities. Economic requirements for WSNs require nodes to be small in size and cheap concerning their production costs. This implies that a physical protection (e.g., TPM [3]) is not an option. Security and trust concepts are often used interchangeably although they have a different meaning.

A common definition of trust is given by Mayer et al. ([4]) where he defines trust as the willingness of a party to be vulnerable to the actions of another party, based on the expectation that the other will perform a particular action important to the trustee. Thus, security is a situation where risks encountered are reduced by special activities to a level of acceptable risk which depends on trust we have on system's functioning.

Although security and trust are key requirements in such scenarios, these challenges for WSNs have not been solved yet [1]. It is argued that cryptography will solve the problem such as well-known methods used in traditional networks. However, this approach is questionable since a node could be picked up and analyzed by the attacker (i.e., extract cryptographic keys from device). Inside attacks, where an attacker has control over one or more already legitimate authorized devices of the network, can be thwarted by a trust reputation system. Byzantine attacks, in which a fraction of nodes are compromised, are another challenge. Several kind of Byzantine attacks are known, such as black hole, flood rushing, wormhole, and replayed routing information ([5]). These attacks work perfectly under the assumption that devices or a set of devices can be captured by attackers and afterward compromised to obtain control over parts of the network or to prevent the sensor nodes from fulfilling their task. Protection against these kind of inside attacks is challenging. For example, authentication and data integrity mechanisms such as cryptography are not able to provide protection, because these mechanisms can not force a sensor node to behave according to the protocol. Awerbuch et al. define in [5] a Byzantine attack as any attack that involves the leaking of authentication secrets so that an adversarial device is indistinguishable from a legitimate one.

To summarize, making one single node secure is nearly impossible due to the limited available computational power and memory as well as cost constraints. But it is possible to improve the security of the whole WSN by using redundancy and incorporate voting. Trust establishment is needed to evaluate the trustworthiness of other nodes.

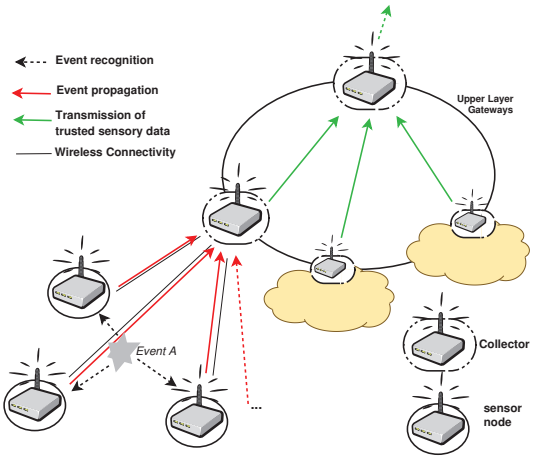



Fig. 2. A WSN with trustworthy data collectors [6].

A generic scenario, as it is shown in Figure 2, is used to analyze the requirements. In this example, a data collector node (gateway) collects sensory data which nodes have reported. The general question is how nodes in a cooperative communication can rely on each other and decide whether and with whom to interact in these uncertain conditions. The data collector has to verify the information, and in the case of acceptance the node publishes the new sensory data to the next hop (here another collector node). In our scenario all nodes associated to a collector node have physical proximity to the central collector node and thus form a local limited monitoring environment. Hereby, the central node is responsible for data collection and verification of data authenticity, integrity, confidentiality, and freshness. Other security aspects may be an option but depend on specific scenario requirements [6]. However, dependent on the sensory environment and the WSN task different scenarios are possible as well. In all scenarios the node continuously receives messages and has to prove how trustworthy and relevant the new information is. E.g., Radmand et al. have summarized in [1] the security issues surrounding WSN with the focus on industrial usage. The following three general requirements are identified to be mandatory to thwart insider attacks on sensor network data.

- **Data Authenticity** is crucial for service correctness. An attacker can feed false information by masquerading as a legitimate node. Thus, the receiver needs

to ensure that the data origin has not been tampered and that data is received from the correct source. In traditional networks a Message Authentication Code (MAC) is used to positively identify the communication source and build a trusted peer connection [1]. But MAC is based on a shared secret key and nodes do not provide a secure data storage or tamper resistant hardware. Thus, we need data authentication that is not only based on cryptography in WSN [7].

- **Data Integrity** ensures that no received message has been altered in transit, thus the message is complete and correct [1]. Authentication methods like MAC are used frequently to address this problem. However, a MAC shared secret key can be stolen or a legitimate node can be compromised. Therefore, different criteria should be considered to find a common trust level which represents the state of trustworthiness of the sending node. Thus, for each kind of a sensor a separate trust level can be calculated.
- **Data Freshness** is important to guarantee that monitored events are transmitted to the network operator within a given time-period. Any disruption or delay may have a negative impact on the operation of the enrolled WSN ([1], [8]). In addition to readily available self-management protocols, we also focus on real-time information exchange. Only trustworthy peers should be used to come to a common decision (i.e. looking of event monitoring like object tracking). On the other side events reported only by a minority of nodes will not be transmitted at once, because in the beginning their calculated trust level will be low. Thus, there is a tradeoff between how trustworthy a message appears and its transmission delay. Concerning the data freshness, the assumed delay is optimized to a minimum.

In addition to the security requirements data freshness, integrity, and authenticity, we focus on the following special sensor network requirements.

- **Network dense** A dense sensor network is needed for safety reasons, thus the number of sensors should be high and every node needs a sufficient number of neighbors within its radio transmission area. A huge number of nodes can provide redundancy to overcome node failures and thus to increase system stability, as well as for security reasons [9]. An unique characteristic of WSN is streamed data from sensed events.
- **Data aggregation** To handle a large amount of nodes in a network of resource-limited nodes the use of data aggregation techniques is required. Thus, nodes have to be clustered and reported sensor values of such a cluster aggregated to one value. With such a technique efficient communication is possible on the one hand, but on the other hand attacking the aggregation node is a high risk. A false data injection (FdI) or inside attacks in general have to be identified early in order to reduce harmful effects on network operations.
- **Power efficiency** The most important special requirement in sensor networks is the need for low power consumption. An additional hardware or sophisticating security protocols do not fulfill this requirement. Because of economic aspects tamper protected hardware or a TPM-chip enabled node is only an option for special purpose nodes, such as gateway nodes [10].
- **Robustness against attacks** Security mechanisms are unavoidable for trustworthy cooperative environments like sensor networks for CPS. But such a security mechanism might be itself target of attacks. Thus, robustness against attacks on the used security mechanism is a hard requirement.

## III. Related Work

Ganeriwal and Srivastava have presented the Reputation-based Framework for high Integrity Sensor Networks (RFSN) in [11]. The calculated reputation per node has an aging effect, giving a positive or negative assessment based on the node's history of behavior. Limited resources prohibit challenging measures to ensure data authentication and cryptography. Therefore, the framework is based on Beta distribution, a mathematical tool to represent and continuously update trust and reputation. Actions are classified as cooperative and non-cooperative (good and bad). To calculate more accurate levels of trust, second-hand information is weighted, and more weight is given to information by reliable nodes. Trust is calculated as an expected value of the reputation. If the trust level value is below a given threshold, the node is defined uncooperative, otherwise it is cooperative. RFSN propagates only positive reputation information and nodes are not able to exchange their bad experience about obviously malicious nodes. First, this is a design choice, in respect to bad mouthing attacks. The problem with this approach is that all nodes have to be active all the time in order to react. This design choice neglects resource limitations and is therefore not usable in real networks. Secondly, additional communication is needed which also shortens the lifetime of the system.

Traditionally, work undertaken on trust is based on successful and unsuccessful transactions between entities (binary), and trust has been modeled in networks from a communication point of view. A unique characteristic of WSN is its data (continues sensed events). Momani has presented a different trust and reputation system for WSN called Gaussian Trust and Reputation System for Sensors Networks (GTRSSN) [12]. The idea behind GTRSSN is that a sensor node will observe neighboring nodes' behavior and calculate a reputation for that node based on the observed data. The GTRSSN trust model defines trust to be a Gaussian probability distribution $T_{i,j} = Prob\{-\epsilon < \theta_{i,j} < +\epsilon\}$ with mean $\mu$ and variance $\sigma^2$. The smaller the error $\theta_{i,j}$ the higher will the trust be. The more spread the error is, and its mean shifting to the left and right of 0, the lower will the trust value be. The trust value $T_{i,j}$ is updated for each time period a message is received from node $j$. Second hand information is used to recalculate $\mu_{i,j}$ and ${\sigma^2}_{i,j}$.

Thus, the GTRSSN reputation-based system calculates a trust value based on sent data of a node and reputations on this value sent by neighboring nodes. This approach has more theoretical than practical uses, because of the power required for reputation communication messages. The additional communication lowers the battery power of the node and secondly

sensor nodes can only send a recommendation message if they have heard the transmission, thus never sleeping. Also software modifications on all devices are needed to adapt the approach. Thus, using this approach for already available sensor network applications is hard to realize.

Multi-Criteria Decision Making (MCDM) is the study of decision making for problems with multiple objectives. It has been developed and widely applied in solving complicated decision problems. One possible method to solve MCDM problems is the Analytic Hierarchy Process (AHP). The principle of AHP is to compute a normalized weight vector from pairwise comparison data.

The AHP alternatives are ranked by aggregating the partial evaluations of each alternative on the points of view in a global evaluation and ordering these global evaluations to obtain a ranking.

Table I summarizes the comparison and shows the pros and cons of the different approaches. Reputation systems, such as RFSN [11] and GTRSSN [12], help peers to recognize the trustworthy peers and avoid the malicious ones. However, these systems might be themselves target of attacks such as unfair rating, bias towards positive rating, quality variants over time etc. Adding any single rating should not influence the score significantly and the system has to be resistant to manipulation of reputation scores. The MCDM-based approach shown by Sridhar in [13] is used for fault-tolerant data aggregation, but lacks flexibility and robustness against attacks. None of the discussed approaches is able to handle data freshness. In CPS a real-time ability is recommended, otherwise real-time decision making may fail. Reputation systems are a theoretical option to secure sensor networks, but mostly these systems are modeled in a distributed fashion. Additional communication to exchange reputations and highly-available sensor nodes are needed, and thus the required power consumption is too high to use these systems in reality.

TABLE I. COMPARISON OF RELATED WORK

| | RFSN[11] | GTRSSN[12] | MCDM[13] | TPM[10] |
|---|---|---|---|---|
| General Requirements | | | | |
| Data freshness | no | no | no | no |
| Data integrity | partly | partly | yes | yes |
| Data authenticity | yes | yes | no | yes |
| Special Requirements | | | | |
| Trust Reputation | yes | yes | partly | yes |
| Power efficiency | poor | poor | good | poor |
| Data aggregation | yes | yes | yes | no |
| Attack robustness | partly | no | no | partly |
| Scalability | poor | poor | poor | good |
| Feasibility | theoretic | theoretic | yes | partly |

## IV. APPROACH

Traditional approaches try to secure individual nodes of a WSN [9]. As already discussed, this is quite difficult. Therefore, we propose another approach by using the inherent redundancy of a WSN to secure WSN clusters of off-the-shelf sensor nodes. Assumptions of our approach are as follows:

- Power is not a limiting factor to cluster heads and gateway nodes (since they have a permanent power supply).
- Clustering of sensor nodes form a network and nodes can belong to more than one virtual cluster.
- Nodes within a cluster can confirm measurements of neighboring nodes [9].
- The minimum cluster size is limited by the Byzantine Fault Tolerance [14].

The Byzantine Generals' problem was first introduced by Lamport et al. in [14]. It describes a decision problem where one Commander in Chief and $n-1$ generals communicate with each other. Goal of a Byzantine protocol is to let the honest generals come to a collective decision (under the assumption that the commander is honest). Lamport et al. have formulated the thesis that for $n$ persons with $k$ betrayers there is a communication protocol that leads to a common trustworthy decision if and only if $n \geq 3k+1$. Dependent on the scenario the parameter $k$ has to be defined by the network operator, representing the maximum amount of permitted malicious devices in the network. Here, in a cluster of four members one malicious node is acceptable. In a network of 13 nodes up to 4 malicious nodes can be detected and isolated according to Lamport's Byzantine protocol.

Within a four member big cluster we use an approach which is described in [9] to identify a possible malicious node and suppress information of this node to the cluster head. The approach is based on a comparison of the steamed values sent by neighboring nodes. In the upper layer the cluster head nodes have to aggregate the values and send the common value to the gateway. Communication of virtual cluster head nodes is internal. So, we do not have to deal with security problems such as sniffing or spoofing. Another security problem is that in an upper layer in-network processing is used to map values to sensor events. Therefore, the gateway level has to be protected in order to have a low false-positive rate. We apply here the Dempster-Shafer (DS) evidence theory. The main advantage of this theory is the ability to represent lack of knowledge to capture the intuitive notion of sensor quality. Thus, DS theory has its unique advantages in handling uncertainty in sensor intrusion analysis, namely, the lack of need for specifying prior probabilities of all events and the ability to combine beliefs from multiple sources of evidence. Therefore, cluster heads of overlapping sensor node clusters transmit aggregated values to a gateway where the DS theory is used for decision making (find and balance disloyal clusters). Besides, DS is already used in intrusion detection systems and has improved network anomaly detection significantly [15].

In our scenario one gateway node represents the cluster head for all seven different clusters. Thus, we have seven virtualized cluster heads represented by one physical device. This concept reduces the costs and enables us to build a trustworthy decision making process. We assume here that the gateway node is protected and not compromised by an attacker. For e.g., the gateway device could be placed such that an attacker could not easily get it or some kind of tamper protection is available. Within the gateway device information is gathered and finally it has to be decided if an event reported by $n$ clusters is correct or not. An easy task would be to make only a simple majority decision in this case. The problem, however, is that an on-off attack would bypass this approach.

When two contradictory pieces of evidence occur, a DS theory can be used to calculate a belief.

We explain the evidence theory with the following example. If we toss a coin with an unknown bias, the probability will still assign 50% for Head and 50% for Tail by the principle of indifference. The DS theory, on the other hand, handles this by assigning 0% belief to Head and Tail and assigning 100% belief to the set Head, Tail, meaning "either Head or Tail". More generally, the DS approach allows for three kinds of answers: yes, no, or unknown, whereas the last option of allowing ignorance makes a big difference in evidential reasoning. In DS theory, a set of disjoint hypotheses of interest, e.g., attack, no-attack, is called a frame of discernment. The basic probability assignment (bpa), distributes the belief over the power set of the frame of discernment ($\theta$) and is defined as:

$$m_\theta : 2^\theta \rightarrow [0, 1] \tag{1}$$

The belief function ($bel(A)$) shows how much confidence we have in that one of the hypotheses contained in $x$ holds (without specifying which). DS has a combination method, with the goal to combine evidence for a hypothesis from multiple independent sources and calculate an overall belief for the hypothesis. In general, we have the following rule of combination known as the Dempster rule.

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B) \tag{2}$$

The belief function is a belief measure of a proposition A, and it sums the mass value of all the non-empty subsets of A. The plausibility function (pl) takes into account all the elements related to A.

$$pl(A) = 1 - bel(\neg A) \tag{3}$$

For the subset $A$, $bel(A)$ and $pl(A)$ represent upper and lower belief bounds, and the interval $[bel(A), pl(A)]$ represents the belief range.

DS theory allows specifying a weight on "unknown" rather than specifying precise probabilities for every possible event in the space. This ability is used to represent the lack of knowledge to capture the intuitive notion of IDS sensor quality (which usually turns out to be imprecisely described), without suffering the non-intuitive effects of aggregation that have been observed by researchers. The nature of unknown matches naturally with how humans interpret sensor alerts. When an event is fired, there is some degree (say 15%) of belief that an event is going on. But this belief is not an 85% belief that an event is not going on. Positively, asserting that an event is not going on after seeing an event, such as an environmental change, is contrary to our expectation. Adopting the simple true and false case to capture the information provided by an event would require us to know the prior probability of events, which is hard if not impossible to obtain. By using DS theory, we can assign 0.15 belief to "event"(true), 0 belief to "no-event" (false), and the 0.85 goes to "Don't know" (true, false). Another consequence of this model of sensor quality is that there will be no conflict among different events. When we do not trust an event, we just say "don't know" whether the hypothesis is true, rather than assert that the hypothesis is false. This will not contradict the fact that we may trust another event which derives the same hypothesis being true.

In general, all evidences for one hypothesis can be combined to $bel(H) = 1-(1-e_1(H))(1-e_2(H))\dots(1-e_n(H))$ if all sensors confirm the hypothesis. For a concurrent hypothesis $H$, such as some sensors report a value above a given threshold and other sensor nodes report a value lower or equal to this threshold, all evidences against ($e^-$) and all ones for the hypothesis ($e^+$) are summarized.

$$e^+(H) = 1 - (1 - e_1(H))(1 - e_2(H))\dots(1 - e_n(H)) \tag{4}$$

The summarized evidences can be combined as follows [16]

$$pro(H) = (e^+ * (1 - e^-))/(1 - e^+ * e^-) \tag{5}$$

$$con(H) = (e^- * (1 - e^+))/(1 - e^+ * e^-) \tag{6}$$

If only one hypothesis is available, such as in our scenario, the DS belief and plausibility functions are given as follows:

$$bel(H) = pro(H_i) \text{ and } pl(H) = 1 - con(H_i) \tag{7}$$

Based on the DS theory we calculate on the gateway device the trustworthiness of a detected event. Here, an event is defined as set if a reported sensor value is higher than a given threshold of value $X$. Furthermore, the value of $X$ depends on WSN application and type of sensor. Instead of a threshold, a more sophisticate rule could be used as described in [6], which has no influence on the presented DS-based decision system.

The process is as follows. All sensors nodes periodically report their values to the gateway device node by node . The gateway device than maps the sensor values to virtual clusters (here four sensors for each cluster), aggregates the reported values for a cluster, and the decision making process is started based on these seven aggregated values. Based on the decision rule, for example if a sensor value is above or under a given threshold, a DS evidence evaluation is used to decide which of the two possible beliefs are more trusted.

The following listing gives an overview about the application of DS and the implementation of our concepts. The variables e1 and e2 represent the evidences $e^+$ and $e^-$, the parameter X stands for the threshold value, and $j$ for the number of the relevant cluster. For the reliability calculation of clusters two methods are possible:

1) We increment the counter variable in each round for cluster $j$ if its reported value is agreed by the majority of the other clusters. The variable divided by the total number of rounds gives the reliability of the cluster. With this method, no additional reliability function is needed but for each cluster rearrangement the counter variables have to be cleared which could result in a short system blackout. Depending on the scenario a short blackout can be acceptable, otherwise this method cannot be used. Furthermore, legal environmental changes are detected very late, because changes are propagated slowly.
2) We use the Gaussian-based approach presented in [9] to calculate the trustworthiness of the cluster based on continuous sensor values. The calculated trust value is used as a reliability value and calculated on the multi-cluster head node. Therefore, even if a cluster mapping is changed, it is possible to quickly use the new calculated reliability values. The problem is

that completely compromised clusters will also report a high trust value, thus an overlapping of cluster members and a reassembling of clusters is used here to overcome this problem.

In the following the second method is used taking the data freshness requirement and system hardening into account.

Listing 1. DS calculation of belief that values are above threshold X

```
initialize variables
% process reports of seven clusters per round
for j=1 to j<=7 step
   if (value[j] > X)
   then
        e1*=(1-(reliability[j]));
        reliability[j]++;
        counter_e1++;
   else
        e2*=(1-(reliability[j]));
        reliability[j]++;
        counter_e2++;
   end if
next
% calculate evidence
if (e1 > 0) then e1=1-e1;
if (e2 > 0) then e2=1-e2;
if (counter_e1 > counter_e2) then
        % belief threshold reached -> 1
        return (e1 * (1-e2)) / (1 - e1 * e2)
else
        % belief threshold not reached -> 0
        return ((1-e1) * e2) / (1 - e1 * e2)
end if
```

## V. Proof of Concept

So far, we have described our approach and have shown parts of our prototypical implementation. In this section we present results of a real implemented 14 nodes system. We have used XBow MicaZ-series off-the-shelf WSN nodes with a standard and unmodified TinyOS example application called OSCILLOSCOPE. This application uses for the gateway node a MicaZ BaseStation node which reports all messages to a Java application. In our test environment only the Java based analysis tool is modified according to the proposed algorithm. The modification enables us to map nodes to seven virtual clusters and to calculate our belief based on the presented DS theory. In Figures 3(a) and 3(b) a typical measurement is shown for a scenario of 14 nodes mapped to 7 clusters with no attacking nodes. Between 120 and 170 cycles a regular environment change happens. The calculated DS belief (values are above the threshold) is correctly set to 0 for this time period. Thus, a regular change of the environment is not identified as an attack on the system.

In a second scenario we have modified 5 out of 14 sensors. These sensor nodes behave maliciously and send values below the threshold value in order to suppress an event alert. Figure 4(b) shows the aggregated values for all seven clusters over time. Here, nodes 1 – 5 behave maliciously, only reporting values below the given threshold. These malicious nodes belong to clusters 1, 2, 3, and 7, where cluster 1 is mapped to nodes 1 – 4 and cluster 2 is mapped to nodes 2 – 6. The malicious sensor node 5 belongs to cluster 3 and the malicious nodes 1 and 2 also to cluster 7. Concerning the Byzantine agreement problem one malicious node out of 4 nodes is acceptable. Therefore, we are using our approach presented in [9] to identify malicious nodes within the clusters and ignore measurements from detected malicious nodes. Thus, cluster 3 reports correct and trustworthy values in contrast to clusters 1, 2, and 7. For theses clusters we can not guarantee that the values are correct or not. We apply the DS theory on every cluster to give an estimation concerning the system trustworthiness.

As shown in Figures 4(a) and 4(b), two misbehaving clusters are identified by our system correctly and the belief function is stable. The implementation needs about 60 rounds until a malicious cluster has been identified because in the start phase nodes have to commit values one after the other first. After about 80 cycles and after a clean start, the system is ready and operative. The little peak at about 125 cycles is explained with the short period where in our measurement cluster 2 reports correct values and therefore enriches his reliability counter. Even in a situation where an attacker controls some nodes from the beginning, we can reduce harmful effects and present a trustworthy decision.

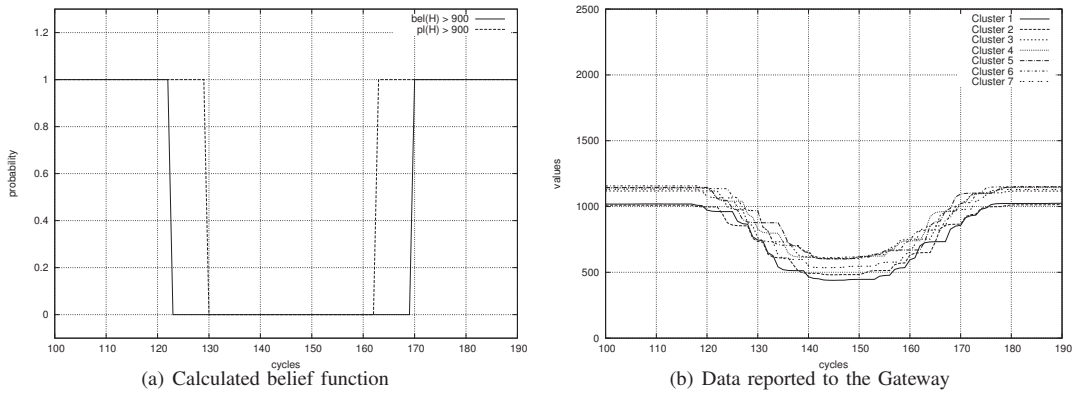
## VI. Assessment

Our solution does not depend on a cryptography or TPM implementation. One single node with such a TPM chip needs on average 58 mA current [3]. A cluster of 13 off-the-shelf nodes w/o TPM needs less than 58 mA current (Figure 5). For such small clusters, power consumption at the gateway is not a problem [10]. With 13 nodes in a static cluster almost 4 malicious nodes are acceptable concerning the Byzantine agreement problem. Thus the system is 4 times harder to attack than a system with only one single trusted node. Thus, a 13 nodes cluster $\widehat{=}$ 1 trustworthy TPM-equipped node under the assumption that not more than 4 disloyal nodes have to cooperate for a successful attack. With a smart cluster node scheduling seven four-node big virtual overlapping clusters out of the 14 available sensor nodes are build.

Figure 5 shows power consumption and the Byzantine failure rate. A power consumption of 14 nodes (XBow MicaZ) is equal to one sensor node with an additional TPM-chip. With 14 nodes the Byzantine failure rate is about 0.69. Thus, concerning Lamport a Byzantine protocol is able to identify 4 malicious nodes [14]. Instead of using a single TPM chip on one node, we use in our approach 14 independent nodes. The achieved power safe can be used to enlarge the lifetime or for further hardening the system. This choice is up to the network operator and depends on the scenario.

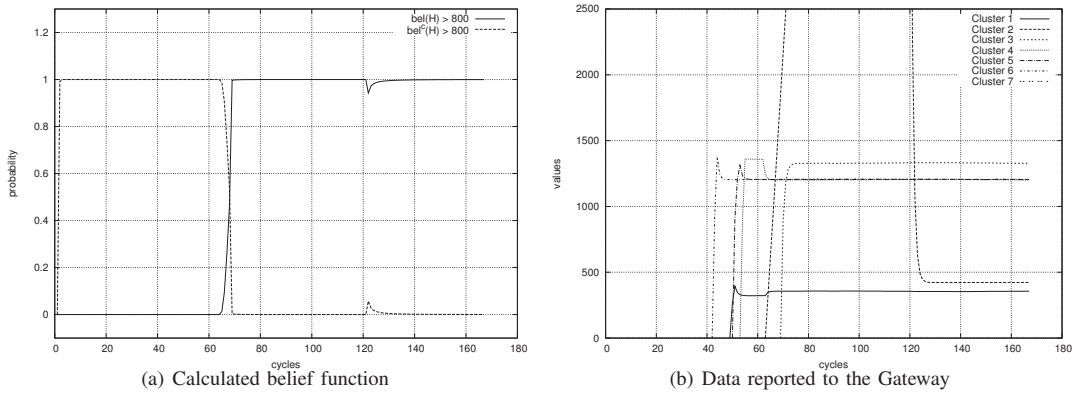
Our concept assumes that the gateway node is highly protected. There already exist approaches to secure special nodes in a sensor network (e.g., [10]). A secure and encrypted communication can be used in addition to our concept; the problem is set in the key distribution and cryptographic key storage on non tamper-proof equipment.

In the presented approach sensors are randomly mapped to clusters. After a certain time, this strategy is reused to establish new sets of clusters. A cluster scheduling is needed to harden the system and thus to reduce the ability of an attacker to take down the system. As mentioned before, we assume that only a small amount of nodes are under the control of an attacker




(a) Calculated belief function (b) Data reported to the Gateway

Fig. 3. Scenario with no malicious nodes and legal environment change




(a) Calculated belief function (b) Data reported to the Gateway

Fig. 4. Scenario with five malicious nodes attacking the network

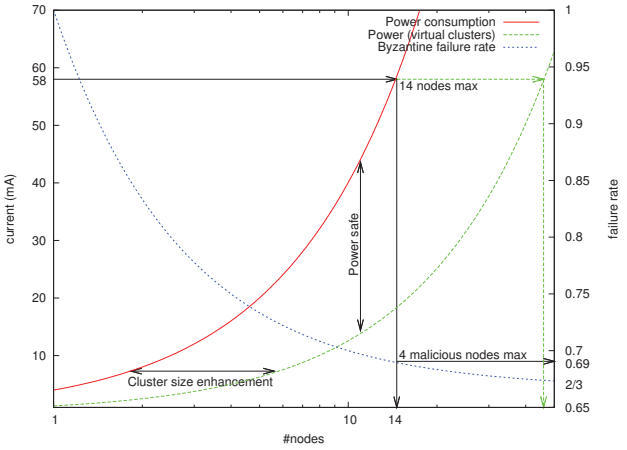



Fig. 5. Assessment: Power Safe vs. System Hardening

and the Byzantine agreement problem gives us an upper bound of acceptable malicious nodes and clusters. With 5 malicious nodes it is possible to have a harmful effect on 4 out of 7 clusters in the worst case. In all other cases only 3 clusters are involved. Even if 5 nodes are under the control of an attacker, the nodes have to cooperate and have to be placed smartly to affect 4 clusters. If the fifth node belongs to one of the already effected clusters, its harmful effect on the system is neglected.

In our proof of concept at least 4 malicious nodes can be identified and isolated, so that these misbehaving nodes do not have an harmful effect on the whole system. To cope also with 5 malicious nodes, a smart node mapping strategy is used. We reassemble the clusters after some time, so also malicious nodes are mapped to new clusters. The idea behind this is that with a reassembling strategy it is possible that for a scenario with 5 malicious nodes the fifth node maybe mapped to a cluster with 2 other malicious nodes. We have $\binom{14}{4} = 1001$ possibilities to choose 4 nodes out of 14 nodes, because we have four overlapping nodes, two of clusters 1, 7 and two of clusters 1, 2. The next 4 nodes out of 10 remaining nodes are mapped to cluster 3, and overlap to cluster 2 and 4. So in the end we have $\binom{14}{4}+\binom{10}{4}+\binom{6}{4}+2 = 1001+210+15+2 = 1228$ possible combinations to map our 14 nodes to 7 clusters. The chance to choose 5 malicious nodes so that at least 4 clusters are involved is $((7*4)+10)/1228 = 0.0309$. Thus, there is only a probability of about 3% that the attacker will choose the right combination of 5 malicious nodes. The probability for the attacker to choose the wrong set of nodes is therefore $1-0.0309 = 0.9691$. So, a rearrangement is beneficial in order to further harden the system. Without a remapping strategy, a smart attacker has a chance of $2/3$ to select the right set of nodes for a successful attack, if the mapping of the cluster has

a static node mapping.

Therefore, trust calculation is centrally processed within the gateway node without communication overhead or time delay. For communication between gateway nodes a real-time enabled communication system presented in [6] is possible. Furthermore, the presented approach is able to resist inside attacks under the assumption that the number of attacking nodes is lower than $1/3$ of all nodes. The trustworthiness of our approach lies in its ability to adapt quickly to new situations by using redundancy of local near sensors. With additional nodes the life-time can be enhanced and a further system hardening is possible, too. Nevertheless, additional security approaches, such as cryptography etc., can be used side-by-side without changes.

## VII. Conclusions and Further Work

Depending on the underlying model, trust management systems are classified into credential-based trust management systems (i.e. based on the identity of a node) or behavior-based trust (i.e. based on the actions of a node). Credential-based trust systems often assume modified hardware such as additional TPM chips. In the paper, the focus lies on behavior-based trust without a modification of hardware or software on sensor nodes. Related behavior-based concepts such as RFSN or GTRSSN are only theoretical concepts. These concepts neglect power consumption and assume that nodes do not sleep. In the presented concept only the software on the of-the-shelf base station is modified. Thus, sensor nodes are still able to sleep and do not waste energy.

Approaches of related areas such as P2P and Ad-Hoc cannot be adopted for WSN due to the difference in the features of these networks. Other security or fault tolerance protocols (e.g., regarding communication) can be used in parallel without modification which is not provided by related concepts. Without a soft real-time ability concerning communication a reputation-based trust management system is vulnerable against message delay attacks. The communication system and the OS scheduler on the base station have been extended to provide real-time ability.

The evaluation shows that 14 of-the-shelf nodes need as much power as one specialized node with an additional TPM chip. In larger networks with hundreds of nodes we have to find 14 dedicated nodes to evaluate the trustworthiness instead of using specialized TPM-equipped nodes. The DS evidence theory can be reused in the aggregation path from clusters to the final gateway. Thus, scalability is not a problem and already available redundancy is not only used for fault tolerance but also for security reasons.

A Gaussian based trust reputation system based on raw measurements is used within a 4-nodes cluster. Therefore, under consideration of the Byzantine thesis one malicious node in this small cluster can be neglected and malicious reports filtered out. On the collector node the DS evidence theory is used to identify malicious behaving clusters consisting of more than one malicious node and is executed for every cluster. An attacker has to succeed in overcoming these two barriers. Any subset of up to (n-1)/3 malicious nodes has no harmful effect on the system and for most subsets although (n-1)/3 + 2 malicious nodes are tolerable.

In the evaluation the power consumption costs for two scenarios are compared. Our approach uses of-the-shelf sensor nodes. Sensor nodes with TPM technology are currently not available on the market or are home-build nodes, thus an economic evaluation which such nodes is not possible.

## Acknowledgment

The authors wish to thank the members of the Chair for Communication Systems and Internet Services at the Universität der Bundeswehr München, headed by Prof.Dr. Gabi Dreo Rodosek, for helpful discussions and valuable comments on previous versions of this paper.

## References

[1] P. Radmand, A. Talevski, S. Petersen, and S. Carlsen, "Taxonomy of Wireless Sensor Network Cyber Security Attacks in the Oil and Gas Industries," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. IEEE, 2010, pp. 949–957.

[2] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. S. Sastry, "Challenges for securing cyber physical systems," in *Workshop on Future Directions in Cyber-physical Systems Security*. DHS, July 2009. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/601.html

[3] M. Kim, Y. Kim, and H. Cho, "Design of Cryptographic Hardware Architecture for Mobile Computing," 2009.

[4] R. Mayer, J. Davis, and F. Schoorman, "An integrative model of organizational trust," *The Academy of Management Review*, vol. 20, no. 3, pp. 709–734, 1995.

[5] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-rotaru, and H. Rubens, "Mitigating Byzantine Attacks in ad hoc Wireless Networks," 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.3.6309

[6] B. Stelte, "An Event Management System for Wireless Sensor Networks," in *IM 2011 - TechSessions*, Dublin, Ireland, May 2011.

[7] ——, "State-of-the-Art Kryptoverfahren für drahtlose Sensornetze – Eine Krypto-Bibliothek für MantisOS," in *Proceedings of GI SIG SIDAR Conference on Information Security (Sicherheit 2010)*, October 2010.

[8] A. Matheus and B. Stelte, "Evidence theory for reputation-based trust in wireless sensor networks," in *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications*, ser. COM.Geo '12. New York, NY, USA: ACM, 2012, pp. 38:1–38:2. [Online]. Available: http://doi.acm.org/10.1145/2345316.2345360

[9] B. Stelte and A. Matheus, "Secure Trust Reputation with Multi-Criteria Decision Making for Wireless Sensor Networks Data Aggregation," in *Sensors, 2011 IEEE*, oct. 2011, pp. 920 –923.

[10] C. Krauß, "Handling insider attacks in wireless sensor networks," Ph.D. dissertation, TU Darmstadt, 2010.

[11] S. Ganeriwal, L. Balzano, and M. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 3, p. 15, 2008.

[12] M. Momani and S. Challa, "GTRSSN: Gaussian Trust and Reputation System for Sensor Networks," in *SCSS (1)*, 2007, pp. 343–347.

[13] P. Sridhar, A. Madni, and M. Jamshidi, "Multi-criteria decision making in sensor networks," *Instrumentation & Measurement Magazine, IEEE*, vol. 11, no. 1, pp. 24–29, 2008.

[14] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, pp. 382–401, July 1982.

[15] Q. Chen and U. Aickelin, "Anomaly detection using the dempster-shafer method," in *International Conference on Data Mining (DMIN2006)*, 2006.

[16] J. Barnett, "Computational methods for a mathematical theory of evidence," in *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2*. Morgan Kaufmann Publishers Inc., 1981, pp. 868–875.