Poster: Evidence Theory for Reputation-based Trust in Wireless Sensor Networks

Björn Stelte and Andreas Matheus
Universität der Bundeswehr München
85577 Neubiberg, Germany
Email: bjoern.stelte,andreas.matheus@unibw.de

***Abstract*—Attacks like fault data injection are not easy to prevent in resource-limited sensor networks. Especially in environments with urgent decision making trustworthy sensor networks are mandatory. Redundancy can be used to detect and isolate malicious behaving nodes and thus to secure the network. The presented approach uses off-the-shelf sensor nodes and is more power efficient than one-single trusted node implementations with TPM technology.**

I. Introduction

A typical Wireless Sensor Network (WSN) consists of hundreds off-the-shelf cheap sensor nodes. Each sensor node is equipped with a power efficient micro-controller, a wireless transmitter, and sensory for environmental monitoring. Applications for WSN can be found in industrial environments, such as monitoring critical infrastructure or habitat monitoring as well as in military scenarios for urgent decision making. In such environments assurance of at least a minimum level of security is mandatory. Thus, trustworthy WSN are needed. Simple securing hardware is difficult due to existing resource limitations in particular power consumption and lack of tamper resistant. But redundancy is an inherent feature of WSN where sensory is overlapping. Today, device redundancy is only used for failure tolerance and not for securing the network. In this paper we will show an approach to use device redundancy in WSN to detect and isolate malicious nodes and thus efficiently protect off-the-shelf WSN.

II. Protected Wired Sensor Networks

E.g., the Open Geospatial Consortium (OGC) efforts defined an architecture as a practical approach for integrating sensors and sensor data into an OGC Sensor Web Enabled-based architecture and disaster decision making such as the "Arctic Climatology Sensor Network Prototype". A trustworthy WSN needs beside secure communication trustworthy components, so every sensor node should be implemented as much secure as possible. Here, integrity and authenticity are needed to establish trust. But making a single node secure is impractical due to low computational power and memory as well as cost constraints. Sensor nodes are not tamper proof nor will be in future [1]. Thus, each sensor node is vulnerable and therefore no node nor its data transmission should be considered secure or trusted. If the network operator can decide if an event message is reported by a loyal sensor node or not – or in other words, if disloyal sensor nodes will have no impact on the operator, then we have a trustworthy sensor network. WSN are typically clustered, thus nodes periodically report current measurements to cluster heads, who send aggregated data to a gateway. In wired sensor networks a usage of TPM is a standard approach, but it needs extra power and is too expensive in WSN environments. In wireless environments, especially WSN, as much power as possible has to be preserved and communication overhead reduced. An analyzation of transmitted values instead of communication behavior can prevent false data injection attacks.

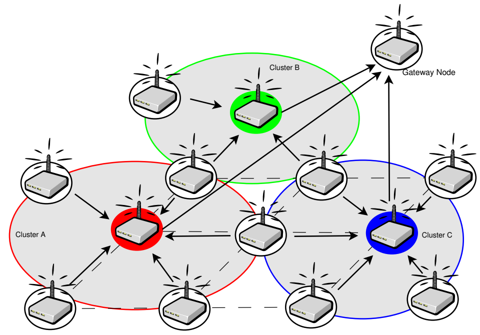

Fig. 1. Overlapping cluster architecture

III. Our Approach

Traditional approaches try to secure individual nodes of a WSN. In our approach we use redundancy to secure WSN clusters of off-the-shelf sensor nodes. We assume that power is not a limited to cluster heads and gateway nodes. Clustering of sensor nodes form a network and nodes can belong to more than one virtual cluster (Figure 1). Nodes within a cluster can confirm measurements of neighboring nodes. The minimum cluster size is limited by the Byzantine Fault Tolerance [3]. Our approach can be characterized as follows:

- Use a **reputation-based trust system** on cluster head nodes to detect malicious nodes.
- Find evidence in measurements to confirm the trust estimation by **pattern matching**.
- Overlapping clusters with separated head nodes use **Dempster-Shafer theory** on gateway nodes for a decision making process.

Detected disloyal nodes are isolated and their measurements ignored (e.g., when data aggregation is used). With off-the-

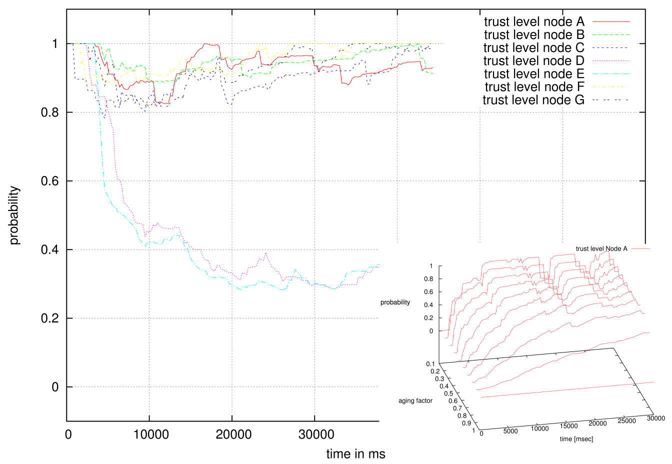

Fig. 2. Calculated trust levels for a sensor cluster.

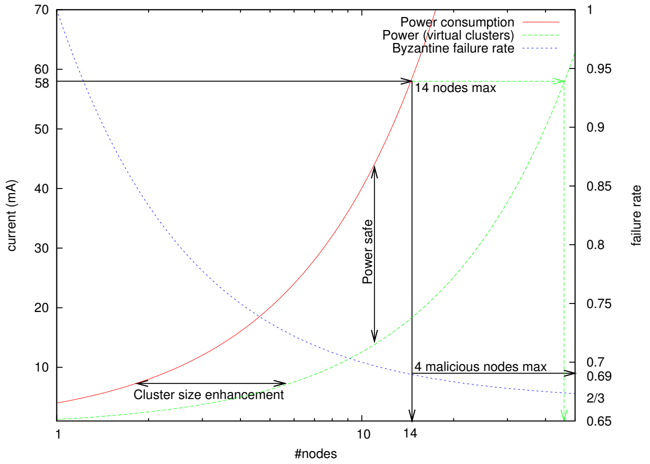

Fig. 3. Power consumption and Byzantine failure rate comparison.

shelf sensor node application and hardware our approach can be used also in already deployed networks.

A. Reputation-based Trust

A process of representing the trustworthiness of one node in the loyalty estimate of another node is used by behavior-based trust management, such as reputation-based systems. Loyalty estimates can be distributed by a trust reputation approach. The assumption is that more than one sensor node monitors an local environment, thus sensor areas are overlapping (Figure 1). The cluster head will observe cluster nodes' behavior and calculate an internal reputation for the nodes based on the observed data. Our trust model defines trust based on probability as shown in the following equation:

$$T_{i,j} = \phi(\frac{\epsilon - \mu_{i,j}}{\sigma}) - \phi(\frac{-\epsilon - \mu_{i,j}}{\sigma}) \tag{1}$$

where ϕ is the cumulative probability distribution of the Normal $N(0,1)$, $\mu_{i,j}$ and $\sigma^2{}_{i,j}$ represent mean and variance. This concept works under the assumption that less than 1/3 of all nodes in a cluster are compromised (Byzantine Agreement Problem [3], $n \geq 3k+1$). Only cluster heads calculate trust values based on transmitted measurements (Figure 2). The advantage of our concept is that no additional communication is needed and that nodes within the cluster do not need to actively wait for transmissions of neighboring nodes to sent a trust reputation (sleep phases are possible). This is a main difference to other reputation-based systems for WSN.

B. Pattern Matching

Since each sensor environment has its own characteristic (like a watermark), this fact can be used to enhance the reputation-based trust calculation. For example, a fluorescent tube generates a significant jitter. Each sensor, monitoring the light intensity of this lamp, reports a measurement data-set where this jitter can be identified. If signature is found, we are willing to trust the corresponding sensor. This means that we will use different trust metric parameters for trust calculation depending on the situation.

C. Dempster-Shafer Theory

The Dempster-Shafer evidence theory is an approach to combine evidence. It is a generalization of Bayesian Theory: Instead of requiring probabilities for each question, belief functions are used. It has the ability to represent lack of knowledge to capture the intuitive notion of sensor quality. Cluster heads of overlapping clusters transmit aggregated values to a gateway which uses Dempster-Shafer theory for decision making (find and balance disloyal clusters).

IV. Conclusions

Our solution doesn't depend on a TPM implementation. One single node with a TPM chip needs in avg. 58 mA current [2]. A cluster of 13 off-the-shelf nodes w/o TPM needs less than 58 mA current (Figure 3). For such small clusters, power consumption at the gateway is not a problem. With 13 nodes in a static cluster 4 malicious nodes are acceptable. Thus the system is 4x harder to attack than a system with only one single trusted node. A 13 nodes cluster $\widehat{=}$ 1 trustworthy TPM-equipped node. More than 4 disloyal nodes have to cooperate for a successful attack. In future work we will use a smart cluster scheduling algorithm for building virtual overlapping clusters. With virtual clusters life-time can be enhanced and a further system hardening is possible.

V. Acknowledgments

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

References

[1] T. Kavitha and D. Sridharan. Security Vulnerabilities in Wireless Sensor Networks: a Survey. *Journal of information Assurance and Security*, 5(1), 2010.

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

[3] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.