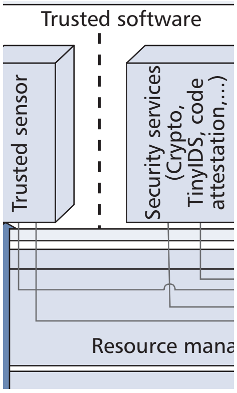
APPLICATION OF WIRELESS SENSOR NETWORKS IN CRITICAL INFRASTRUCTURE PROTECTION: CHALLENGES AND DESIGN OPTIONS

LEVENTE BUTTYÁN, BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
DENNIS GESSNER, SIRRIX AG
ALBAN HESSLER, NEC EUROPE
PETER LANGENDOERFER, IHP MICROELECTRONICS



The authors discuss the challenges and potential solutions to achieve dependability of WSNs taking into account accidental failures as well as intentional attacks. We inspect the whole system starting from individual sensor nodes via the protocol stack to the middleware layer above.

ABSTRACT

The protection of critical infrastructures provides an interesting application area for wireless sensor networks. Threats such as natural catastrophes, criminal or terrorist attacks against CIs are increasingly reported. The large-scale nature of CIs requires a scalable and low-cost technology for improving CI monitoring and surveillance. WSNs are a promising candidate to fulfill these requirements, but if the WSN becomes part of the CI in order to improve its reliability, then the dependability of the WSN itself needs to be significantly improved first. In this article we discuss the challenges and potential solutions to achieve dependability of WSNs taking into account accidental failures as well as intentional attacks. We inspect the whole system starting from individual sensor nodes via the protocol stack to the middleware layer above.

INTRODUCTION

Critical infrastructures (CIs), such as transportation and energy distribution networks, are essential for our society, and therefore are expected to be available 365 days a year, 24 hours a day. Unfortunately, accidental failures of and/or intentional attacks on a CI may result in the disruption of its services. Failures may be caused by bad weather conditions or natural disasters, while attacks may range from mere vandalism to terrorist activities. Real world examples include the failure of the energy distribution network due to heavy snowing in Münsterland, northwestern Germany, in 2005, and the disruption of telecommunication services by intentionally cutting some optical fibers in Morgan Hill, Northern California, in April 2009. These and similar examples clearly show how vulnerable CIs are, and hence, how CI protection is important in our society today.

Critical infrastructure protection (CIP) requires monitoring mechanisms that enable us to detect failures and attacks as early as possible. Since many CIs have a large geographical span, CIP needs monitoring mechanisms that scale well. In this context wireless sensor networks (WSNs) arise naturally as a potential solution. In particular, WSNs can be relatively easily deployed on a large scale, and as they are normally built from low-cost devices, they can provide the monitoring service in a cost-efficient manner since they do not require additional infrastructure. In addition, the distributed nature of a WSN increases the survivability of the network in critical situations, because a large-scale WSN is much less likely to be affected in its entirety by failures or attacks. In very critical situations WSNs may still provide sufficient information about the CI that help the operator prevent further damage and begin the recovery process.

It must be clear, however, that the usefulness of WSNs for CIP is primarily determined by the dependability of the WSN itself. A WSN that fails reporting a faulty condition prevents the CI operator from carrying out the appropriate maintenance that may fix the problem before its consequences affect the CI. System aspects, such as redundancy, integrity, real-time behavior, as well as security and availability are essential requirements to make the WSN, and hence the monitoring services it provides, dependable.

In the WSAN4CIP Project,[1] we work on the application of WSN technology in CIP, and for this reason, we develop mechanisms that increase the dependability of WSNs. In the WSAN4CIP project, and even more in this article, we are focusing on security issues in WSNs. Overall, our

[1] *The WSAN4CIP Project is a European project in which 11 partners from 7 countries are collaborating. The project started in January 2009, and its duration is three years. More information about the project is available at http://www.wsan4cip.eu/*

1536-1284/10/$25.00 © 2010 IEEE

goal by improving the dependability of WSNs in CIP applications is to ensure that failures of and attacks on the WSN monitoring a CI have a minimal impact on the CI itself. Failures may happen and attacks may be targeted at any layer of the WSN architecture from the node hardware and operating system, through the networking protocol stack, up to the middleware and service layers. Therefore, our approach in the WSAN4CIP Project is to address the problem of dependability at *all* layers of the architecture.

In this article we give an overview of the security challenges we face and the approaches we have already adopted to ensure dependability at three layers in WSNs: the node architecture, the networking protocols, and the service layer on top. The next section introduces three complementary approaches to increase the resistance of individual nodes against failures and attacks: intrusion detection and prevention, separation of critical parts of the system, and software-based remote code attestation. The following section is concerned with the dependability of the networking layer, discussing challenges related to the creation of robust network topologies, the secure and reliable transport of data, and the prevention of traffic analysis. Finally, we describe our approach to provide a dependable and persistent distributed data storage service within the network that is network-failure- and attack-resistant.

Hardened Sensor Node Architecture

In order to ensure that a WSN is working correctly, it is essential to guarantee that all individual nodes are working properly (i.e., that they have not been compromised). In the WSAN4CIP Project we are investigating three different means to prevent malicious attackers from successfully manipulating individual nodes. First, we propose to use intrusion detection and prevention techniques adapted to the WSN environment. Second, we are studying the introduction of a micro-kernel approach into the world of sensor node operating systems in order to support different levels of security within a single node. Finally, we are investigating methods of code attestation to verify whether or not the code deployed on a sensor node is still unchanged.

TinyIDS

Detecting and preventing network attacks on WSNs is an essential prerequisite for improving their reliability. Without such a detection, nodes or even the whole network might be compromised without being noticed. Standard intrusion detection systems, such as the well-known SNORT system, cannot be used on sensor nodes due to their constraint resources. SNORT requires about 115 Mbytes storage just to hold its rule set and another 6 Mbytes for its executable. Thus, a simple port to sensor nodes is infeasible.

Instead, in the WSAN4CIP Project we are investigating an approach that allows designing an intrusion detection system (IDS) tailored for specific WSN application scenarios. In our approach, we are exploiting the fact that when the WSN is used for surveillance or monitoring, the behavior of all subsystems and even individual nodes is predefined (e.g., it is usually defined a priori when measurements will be taken, sent, etc.). Therefore, the core of our approach is to use such a description of the system and its components to derive the expected correct network behavior. The assumption is that any other traffic can be considered to be malicious. The definition of the correct behavior can be stored on the sensor nodes themselves (even a complex rule such as detecting a fire and reacting appropriately can be reflected by just 80 characters). To detect attacks each wireless sensor node needs to identify deviations from the expected behavior. This activity is required only during the predefined activity periods (e.g., about every 30 s in one of our scenarios).

To avoid false positives, some deviations need to be reflected in the rule set, for example, by defining guard intervals before and after the predicted transmission time. Another issue to be considered is alarm messages, which might be sent at arbitrary times. Such messages need to be delivered, and additional measurements are required in order to detect whether the alarm was correct or not.

We are currently developing a specification language which allows the definition of the system's behavior. This includes allowed deviations such as retransmissions. We define potential countermeasures for situations in which the correct behavior and potential attacks cannot be identified directly. In order to support the mapping from the system description to a real life supervision application, we also develop a code generation tool.

We define potential countermeasures for situations in which the correct behavior and potential attacks cannot be identified directly. In order to support the mapping from the system description to a real life supervision application, we also develop a code generation tool.

Operating System Issues

Existing sensor platforms do not provide standard security mechanisms, such as memory management units, which can be used to protect different applications from each other. Even if these means are provided, an isolation of security relevant operations, such as cryptographic operations, is not feasible without additional mechanisms. In order to improve the security of the sensor node at the operating system (OS) level, we investigate the idea of introducing a micro-kernel-based architecture for sensor nodes.

The general idea of our architecture is to establish various compartments on a single sensor node platform. Compartments are separated from each other in a way that the information flow between the compartments can be controlled, and is restricted to predefined channels. Each compartment can have its own security policy. These features allow protection of sensitive data, such as cryptographic keys, on the sensor node even if parts of the sensor node are compromised. In the WSAN4CIP Project we aim at implementing this micro-kernel architecture. Our implementation will support the separation of cryptographic operations, and it will allow the secure coexistence of two OSs, virtualized on a single sensor node.

Figure 1 illustrates how each compartment is isolated on the sensor node, and it conveys our idea on how we can use a small Trusted Computing Base (TCB) of a micro-kernel-based system. It uses virtualization techniques and ideas taken from the field of Trusted Computing, such as code attestation, to build a secure sensor node.

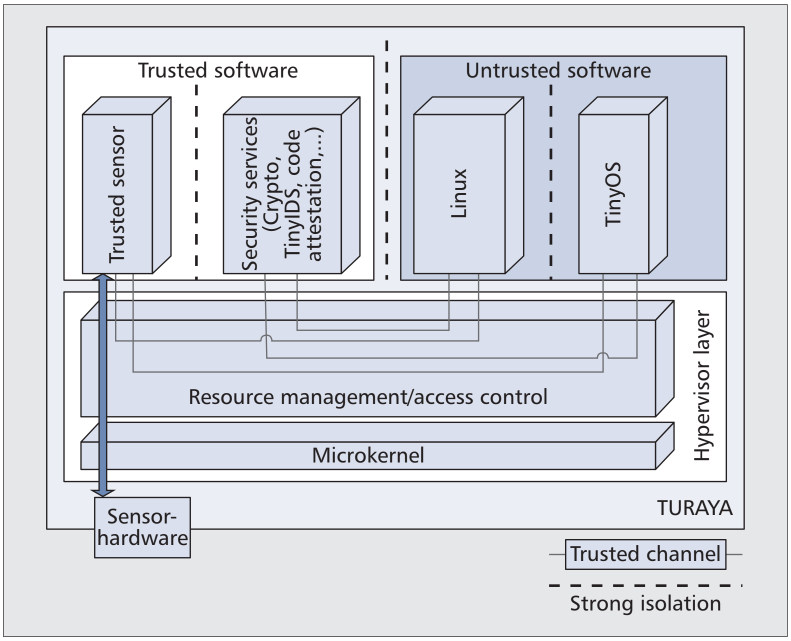


Figure 1. *Illustration of the WSAN4CIP secure software architecture. It allows different operating systems to be run on top of a single sensor node in different security compartments. The resource management and access control layer provides isolation of compartments. Access to the isolated security functions is provided only through this layer as indicated by the arrow going from the TinyOS protocol stack to the security compartment.*

Code Attestation

Preventing nodes from being compromised is difficult; it is therefore desirable to detect compromised nodes in order to isolate them from the network. This is usually performed through code attestation, whereby the base station verifies that each of the nodes is still running the initial application and hence has not been compromised. We are interested in software-based attestation techniques, because they require neither dedicated hardware nor physical access to the device. Existing techniques are based on a challenge-response paradigm, where the verifier (usually the base station) challenges a prover (a sensor node) to compute a checksum of its program memory. To prevent replay or precomputation attacks, the challenge contains a nonce to be included in the checksum computation. Since the verifier is assumed to know the exact memory contents and hardware configuration of the prover, it can compute the expected response, and compare it with the received one.

One of our partners in the WSAN4CIP Project analyzed several existing software-based code attestation techniques, discovered weaknesses in all of them, and proposed practical attacks against existing software-based attestation schemes [1]. Most of the attacks use code compression to free memory space, which can be used to host some malicious code. At attestation time, the malicious code can decompress the original code on the fly, retrieve the original content of the program memory, and succeed in the attestation. Such malicious code can hide itself in non-executable memory locations and use the ROP technique called *return-oriented programming* (ROP) [2] to load itself into the program memory when needed.

Therefore, a dependable code attestation scheme must ensure that the attested device is running the original code in its program memory while it is not storing any other code in any of its other memories. In order to ensure this, our approach to solving the code attestation problem consists of offloading all stored data to the verifier, compressing the content of the program memory, filling the program memory space freed by compression and the unused data memory with fresh random data received from the verifier, and computing the checksum over *all* of the memories of the device. If the attestation is successful, the boot-loader decompresses the application and reboots the device, which returns to operational mode. More details of this solution and on attacks against existing code attestation schemes are available in [3].

Dependable Networking Protocols

Besides the dependability of the nodes themselves, the networking protocols the nodes use must also be dependable. WSNs use wireless communications, and it is well known that wireless channels are more vulnerable to environmental noise, and hence less reliable. They are also vulnerable to attacks, such as jamming, injection of forged data, eavesdropping, and replay of communications. In this section we give an overview of the challenges and problems addressed in the WSAN4CIP Project related to the deployment of the nodes, the reliable transport of data, and the concealment of the network hierarchy.

Node Deployment

While many research papers assume that in sensor networks, the nodes are deployed in a random manner (e.g., thrown out of an aircraft), we believe that in the majority of civilian applications, and particularly in CIP applications, the sensor nodes should be deployed systematically. This may be mandated by the specific geography of the CI, and the specific constraints and requirements of the CI monitoring application, and it may also make optimizations possible in terms of maximizing sensing coverage and network lifetime. Besides these basic objectives, systematic node placement can help increase the fault tolerance of the network and its resistance to jamming attacks.

The placement of the nodes and their radio transmission ranges determine the network topology. In order to compare different potential node deployment strategies, one needs a metric to measure the quality of a resulting topology, where quality means resistance to random failures and intentional attacks. The most widely used robustness metric in the sensor network literature is the degree of connectivity of the topology graph. As failures and attacks can affect both nodes and links, both the concept of vertex connectivity and that of edge connectivity are relevant. The vertex/edge connectivity of a graph is the size of its smallest vertex/edge cut, and, intuitively, the larger the connectivity is, the more dependable the topology is.

The connectivity (vertex and edge) based metrics are appealing because they are intuitive and they can be computed efficiently. Yet, they fail to capture some important aspects. In particular, they do not shed light on how a k-connected topology fails, when more than k nodes/links are destroyed.

As an illustrative example, consider Fig. 2. The graph on the left is 2-connected, while the graph on the right is only 1-connected. However, when an attacker can remove two nodes from these graphs, the left graph completely falls apart, while the other one still has a large connected component.

In the WSAN4CIP Project we use the notion of *graph strength* [4] to measure the resistance of a network topology against jamming attacks and link failures. With a minimal modification of the graph that represents the topology, this notion can also be used to measure resistance against node destruction attacks and node failures. Informally, the strength of a graph is the minimum ratio between the total cost of edges removed and the total value of nodes that become disconnected from a designated set of nodes (that can model the base stations in WSNs) as a result of the edge removal. Thus, intuitively, the graph is stronger if more edges need to be removed to disconnect the same amount of nodes, or if fewer nodes get disconnected by removing the same amount of edges. The strength of a graph can be computed efficiently, and it has other desirable properties (e.g., monotonicity). Based on this metric, we can translate node deployment problems into optimization problems that aim at maximizing the strength of a topology under certain constraints. For instance, we can answer questions such as this: if there is an upper bound on the number of gateways allowed, which nodes should be gateways such that the resulting network has maximal strength?

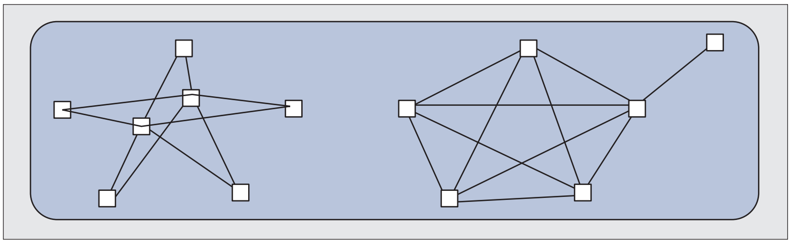


Figure 2. *The notion of* k*-connectivity fails to capture what happens with the topology graph when more than* k *nodes are removed. The left graph is 2-connected, while the right hand side graph is 1-connected. However, when an attacker can remove 2 nodes from each of these graphs, the left graph completely falls apart, while the one on the right still has a large connected component.*

Dependable Protocols

A robust network topology is a good start, but not sufficient: we also need dependable routing and transport protocols that run on that network topology. While reliability and security of routing protocols have been extensively studied in the past, transport protocols have received less attention.

Note that simple monitoring applications may ensure end-to-end reliability by mechanisms integrated in the applications themselves, and they do not necessarily need a dedicated transport protocol. However, in the WSAN4CIP Project, we also consider scenarios where transmission of video streams from small camera equipped sensors could be triggered by some events (e.g., physical intrusions detected by motion sensors). We believe that such video surveillance mechanisms can be useful in CIP in general. Video streaming over a WSN, however, induces transport problems (e.g., congestion control and real-time requirements) that can be more conveniently handled in a dedicated transport protocol below the application layer. A number of transport protocols specifically designed for WSNs have been proposed in the literature with reliability and energy efficiency as their main design criteria. Interestingly, despite the fact that WSNs are often envisioned to operate in hostile environments, none of the proposed transport protocols address security issues. As a consequence, all known WSN transport protocols fail to provide reliability and to ensure energy efficiency in a hostile environment.

In particular, both positive acknowledgment (ACK) and negative acknowledgment (NACK) based protocols are vulnerable to control packet forgery attacks: a forged ACK may create the false impression that the acknowledged data packet has been delivered, while a forged NACK triggers unnecessary retransmissions. In addition, if a single control packet can refer to multiple data packets, the above problems are aggravated, and if a protocol combines the ACK and NACK approaches (e.g., selective ACK schemes), it inherits the problems of both.

To prevent control packet forgery, control packets must be authenticated. As many transport protocols require intermediate nodes on a path to cache data packets until they can be sure that they have been delivered, control packets must be authenticated such that these intermediate nodes can verify them. This means that we need to use a broadcast authentication scheme. The standard solution would be a digital signature scheme, but it is prohibitively expensive in terms of computational resources in WSNs. For this reason, in the WSAN4CIP Project, we are developing specific ACK and NACK authentication mechanisms using only symmetric key bryptographic techniques.

Protection against Traffic Analysis

Traffic analysis aiming at the identification of nodes that play special roles in the network can be exploited by an adversary to increase the efficiency of an attack against the network. For instance, a larger amount of damage can be caused by disabling base stations or cluster head nodes than by destroying ordinary sensor nodes. For this reason, techniques to prevent traffic analysis and hide the role of the nodes in the network can be beneficial.

In this context, anonymous routing protocols that hide the target of packets from the forwarding nodes and from external eavesdroppers that can analyze traffic patterns are very relevant. Such a protocol is proposed, for instance, in [5], where a suite of de-correlation countermeasures are applied including hop-by-hop re-encryption and random delay of packets, as well as mechanisms that introduce randomness into the paths followed by the packets.

However, the problem of preventing the localization of cluster head nodes requires more than an anonymous routing protocol: one must also ensure that the identity of the cluster head is not leaked by the cluster head election protocol. In order to achieve this, messages of the protocol must be encrypted such that external eavesdroppers cannot recognize cluster head announcements, and the protocol must be designed in a way that the frequency and timing

The problem of preventing the localization of cluster head nodes requires more than an anonymous routing protocol: one must also ensure that the identity of the cluster head is not leaked by the cluster head election protocol.

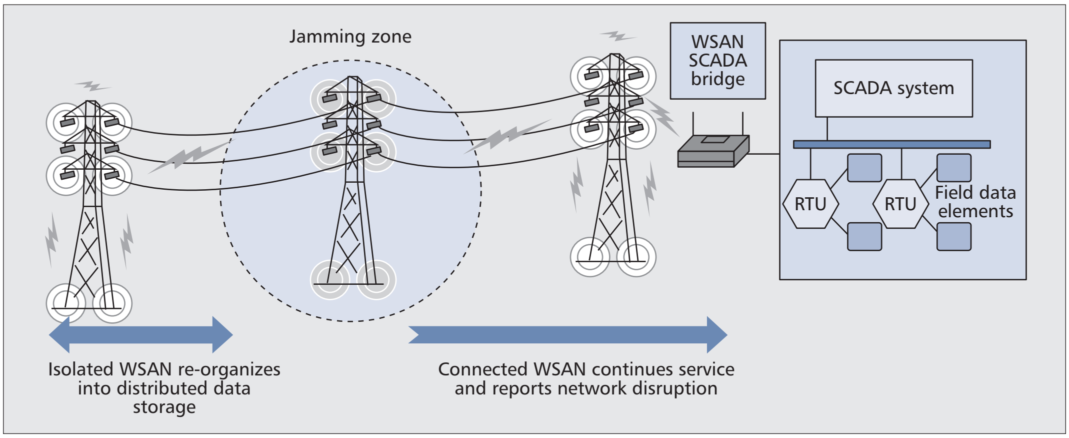

Jamming zone
WSAN SCADA bridge
SCADA system
RTU
RTU
Field data elements
Isolated WSAN re-organizes into distributed data storage
Connected WSAN continues service and reports network disruption

Figure 3. *Illustration of the WSAN4CIP approach to resist to jamming attacks. When an attack takes place and results in network partition, service protection techniques ensure the continuation of the data collection. The collected data is stored and aggregated locally, and reported to the Supervisory Control And Data Acquisition (SCADA) system monitoring the CI after the attack.*

of transmissions do not reveal which nodes are the cluster head candidates. Our first attempt to design such a privacy preserving cluster head election protocol is presented in [6]. That protocol, however, is vulnerable to insider attacks (i.e., attacks mounted by compromised nodes that participate in the cluster head election). Our current research effort in the WSAN4CIP Project is targeted toward the design of a private cluster head election protocol which ensures that the identity of the cluster head remains hidden even from the cluster members (it is sufficient for members to know how to route messages to the cluster head), because in that case, compromising a node would result in no useful information for the attacker.

Dependable Services

Deployed WSNs face a fluctuation of unfortunate factors, such as hardware failures or poor wireless medium, which directly impact their capability to deliver the required service. The challenge at hand is to build applications that, despite those failures, provide *some* service rather than halting completely. This is the property of failing gracefully. Ideally, the performance of the service should decline in relation to the severity of the failure, and restore to full performance as soon as the disruption ends. The problem is, how can we build services that literally bend but do not break?

Similar to the concept of *security in depth*, we want to provide dependability in multiple layers so that if a layer fails, the next one can take over. This approach offers protection diversity at relatively low cost. Based on our threat analysis, we identified jamming as one of the most serious threats to the applicability of WSNs in the CIP context. On one hand, jamming requires little knowledge and resources from the attacker; on the other hand, it can bring down large part of the WSN without proper countermeasures. We show that service protection can provide a technical *last stand* against jamming when all other countermeasures failed.

Consider a typical CIP application, where the WSN reports periodically various parameters of the CI to the Supervisory Control And Data Acquisition (SCADA) system of the infrastructure operator. We assume that a jamming attack is happening, which disconnects a part of the network and all the jamming countermeasures at both the physical and routing layers, failed to establish connectivity, as shown in Fig. 3. In this case there is no means to provide the original service anymore. Once the attack is recognized by the WSN, there are two possible strategies: put the nodes into sleep mode until the attack stops, or continue monitoring and cache the collected data within the WSN, in order to provide it to the SCADA system later on.

To carry on with the data collection, we observe many challenges for constructing a dependable distributed storage based on coding. We present here a relevant subset of the requirements.

Persistency: The distributed database must be fault-tolerant, and thus resilient to node failures. Simple replication and storage of measured data is neither memory-efficient nor dependable: a few node failures might already yield partial data losses. Recent works, such as Reed-Solomon or Fountain codes [7, 8], promote linear erasure codes to ensure a better fault-tolerance behavior of the distributed storage for a same amount of memory. We adopt this approach, however, these schemes generally follow a random fault model. Thus, there is a need to consider more realistic failure patterns, taking into account the topology for example.

Security: Integrity of the retrieved data is a major requirement. Computation-expensive cryptographic integrity schemes are available, e.g., based on homomorphic signatures, but they are not applicable in WSNs. Our approach is based on a non-cryptographic solution, described in [9], which exploits the linear properties of the coding, and also the redundancy and diversity of distributed storage. If the storage keeps more equations than unknown variables, an over-determined linear system is built. Hence, the verifier can retrieve additional codewords from the network and use the extra information to detect and correct corrupted data.

Situation awareness: Distributed event detection and voting schemes are necessary to detect network disruptions and trigger a reaction, such as re-organization into a distributed data storage on the isolated side. Those network state fluctuations should also be reflected on the SCADA system.

Lifetime: The storage service is limited by the amount of available memory on the node. A CIP application typically needs to measure a few integer variables every 30 s for each node, hence a rough estimate for the distributed storage service lifetime is 8 h for 64 kbytes of flash memory. Longer times can be achieved by sacrificing either precision or redundancy.

In addition to addressing the above issues, we also intend to develop an energy management middleware. The nodes will gather performance metrics from different layers of the network stack and exchange information with their neighbors in order to infer a quality of service (QoS) state. In our jamming scenario the interference could be measured at the physical and medium access control (MAC) layers (low signal-to-noise ratio [SNR], numerous corrupted packets); thus, the management middleware would make the decision to set the nodes directly under attack to a prolonged sleep cycle until the radio interference ceases.

Conclusions

Wireless sensor networks are built out of low cost devices, do not require additional infrastructure, and are self-healing. These properties are prerequisites when planning for monitoring of large scale critical infrastructures such as energy distribution networks. Reliability is a key requirement for WSNs used in critical infrastructure protection; however, the level of reliability also depends on system properties such as security, energy consumption, and radio connectivity. Therefore, in the WSAN4CIP Project, we try to take all these issues into account, with a special emphasis on security issues.

In this article we have identified scientific challenges with respect to security and dependability that need to be solved to make WSNs a ready to use technology for CIP. In order to cope with those challenges we apply a vertical approach, where we do not focus on a single layer or subsystem, but consider the whole system in all its facets. We are currently working on implementations of the concepts introduced here. Testbeds using parts of an electricity distribution network in Portugal and a drinking water pipeline in Germany were rolled out in summer 2010. In these testbeds we can run many of our security and dependability solutions, and test them under realistic conditions.

> We are currently working on implementations of the concepts introduced here. Testbeds using parts of an electricity distribution network in Portugal and a drinking water pipeline in Germany were rolled out in summer 2010.

There are still open challenges when applying WSNs for CIP (e.g., tamper resistance and realtime constraints). The latter has been touched slightly in the WSAN4CIP Project by investigating low duty cycle protocols for WSNs, which allow for minimum end-to-end communication delays. The former problem requires changes to the current hardware design and is being addressed by some partners in the European project TAMPRES starting in October 2010.

Acknowledgments

The authors are work package leaders and the project manager of the WSAN4CIP Project, and they are thankful to all participants of the project for their contributions to this article. All contributing participants certainly deserve to be on the author list; however, for practical reasons, we could not include all of them.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 225186.

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 use of the information is at the sole risk and liability of the user.

References

[1] C. Castelluccia *et al.*, "On the Difficulty of Software-Based Attestation of Embedded Devices," *Proc. ACM CCS*, 2009.

[2] E. Buchanan *et al.*, "When Good Instructions go Bad: Generalizing Return-Oriented Programming to RISC," *Proc. ACM CCS*, 2008.

[3] A. Francillon, *Attacking and Protecting Constrained Embedded Systems from Control Flow Attacks*, Ph.D. dissertation, Institut Polytechnique de Grenoble, 2009.

[4] W. H. Cunningham, "Optimal Attack and Reinforcement of a Network," *J. ACM*, vol. 32, no. 3, 1985, pp. 549–61.

[5] J. Deng, R. Han, and S. Mishra, "Decorrelating Wireless Sensor Network Traffic to Inhibit Traffic Analysis Attacks," *Pervasive Mobile Comp.*, vol. 2, no. 2, 2006, pp. 159–86.

[6] L. Buttyán and T. Holczer, "Private Cluster Head Election in Wireless Sensor Networks," *Proc. IEEE WSNS*, 2009.

[7] Y. Lin, B. Liang, and B. Li, "Data Persistence in Large-Scale Sensor Networks with Decentralized Fountain Codes," *IEEE INFOCOM*, 2007, pp. 1658–66.

[8] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized Erasure Codes for Distributed Networked Storage," *IEEE/ACM Trans. Net.*, vol. 14, no. SI, 2006, pp. 2809–16.

[9] L. Buttyán, L. Czap, and I. Vajda, "Pollution Attack Defense for Coding Based Sensor Storage," *Proc. IEEE Int'l. Conf. Sens. Net., Ubiquitous, and Trustworthy Comp.*, 2010.

Biographies

LEVENTE BUTTYÁN (Buttyán@crysys.hu) received an M.Sc. degree in computer science from Budapest University of Technology and Economics (BME) in 1995 and earned a 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 an 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 and privacy enhancing mechanisms for wireless networked embedded systems. Within the WSAN4CIP Project, he leads the work package on "Dependable Networking."

DENNIS GESSNER (d.gessner@sirrix.com) studied computer science at TU Darmstadt and FH Gelsenkirchen, Germany, and joined Sirrix AG in 2007. His professional record includes IT-security positions in pharmaceutics, banking, and research laboratories. Currently, he focuses on operating system security, wireless sensor network security, and trusted computing. Within the WSAN4CIP Project, he is leader of the work package on "Node Protection."

ALBAN HESSLER (Alban.Hessler@nw.neclab.eu) received his M.Sc. degree in communication systems from EPFL in 2006 with a specialization in information and communication security. He thereafter joined NEC where he works on security solutions for wireless systems. Since then he has participated in European projects UbiSec&Sens, SENSEI, and WSAN4CIP. Within the WSAN4CIP Project, he is leader of the work package on "Dependable Services."

PETER LANGENDOERFER (langendoerfer@ihp-microelectronics.com) holds a diploma and a doctorate degree in computer science. Since 2000 he has been with the IHP in Frankfurt (Oder). There he is team leader of the wireless sensor network group. He has published more than 80 refereed technical articles and filed seven patents in the security/privacy area. He is the technical manager of the WSN4CIP Project. His research interests include wireless communication, especially privacy and security issues.