An Engineering Approach for Secure and Safe Wireless Sensor and Actuator Networks for Industrial Automation Systems

Steffen Peter and Oliver Stecklina and Peter Langendoerfer
IHP GmbH
Im Technologiepark 25
15236 Frankfurt (Oder); Germany
{peter,stecklina,langend}@ihp-microelectronics.com

Abstract

Wireless communication and smart sensors and actuators pose means to sustainably improve automation technology. Unfortunately they also cause an abundance of new challenges regarding security and safety of the system. After introducing the security concepts, this paper discusses an engineering methodology to cope with security requirements in context of industrial automation. Two practical examples demonstrate how the solutions even for pretty similar scenarios can differ significantly. The proposed development flow promises a reliable objective engineering of proper system solutions. Key concepts of the flow are a holistic goal description and an iterative composition algorithm that inherently applies and extends existing knowledge.

1 Introduction/Motivation

From the aspect of security engineering two relatively new tendencies in the domain of automation systems are extremely relevant. The first is that more and more fabrication sites are connected to each other via public accessible networks such as the Internet. This approach is motivated by reducing cost for monitoring with a centralized control center. The result is that the formerly isolated fabrication networks are now accessible from everywhere in the world, and by that all Internet-based attacks can be run against fabrication networks. The second tendency is to use wireless communication to a larger intent than up to now. The idea here is to allow more flexible set-ups of manufacturing sites and reduce cost for monitoring of difficult-to-reach devices. Using wireless communication has a similar effect as connecting the fabrication network to the Internet, i.e. the system becomes accessible from outside. From a security engineering approach the exchange of wired communication by wireless is much more severe than going for Internet access. This is due to the fact that wireless connections can be accessed from any position within the transmission range of the used wireless technology. Thus, potential attackers are no longer forced to enter the fabrication network at a well defined entry point - as it is for wired Internet connections. Such entry points are normally powerful machines running strict firewalls. In contrast to those machines the new entry points might even be small sensor nodes, which have limited energy resources, limited processing power etc. Deploying standard protection means on sensor nodes might for example increase the processing time that much that dependability constraints will be violated. So, a straight forward re-use of those concepts on sensor nodes is infeasible.

When designing new security solutions for automation networks it must be ensured that the core functionality i.e. controlling a manufacturing site is not influenced by the security solution. This means constraints such as dependability issues and the current set-up of the system - consisting of software, protocols etc.- need to be taken into account. Especially the latter is difficult to obey while engineering a system since a lot of information is not explicitly modeled but must be inferred. We reflect this by introducing an additional engineering constraint into our semi-automatic approach, which we call *environment*.

The contribution of this paper is the introduction of a holistic but still easy to implement approach which allows engineering security solutions for automation networks. Our approach considers formerly not modeled constraints (i.e. environment) and dependability issues as well as the idea of economically secure systems. By this term we denote the fact that a security solution must ensure that the cost of an attacker to break the security solution is higher than his/her potential benefit. We use a real life examples from the RealFlex project[8] to introduce the security engineering challenges and to illustrate our own solution.

The rest of this paper is structured as follows. Sections 2 we provide a fundamental overview of information security and introduce our examples used throughout the rest of the paper. Our security engineering methodology is presented in section 3. In section 4 we map our approach onto the previously introduced used cases. Then we present related work. The paper concludes with a short summary and an outlook on future work.

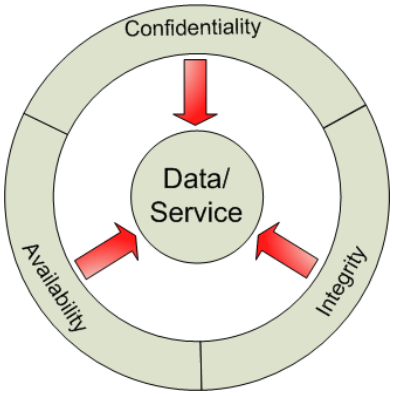

Figure 1. CIA triad: Confidentiality, Integrity and Availability are three key security principles in any kind of information system

2 General Security Terms

In this section we first introduce a general overview of information security, which we will focus on in the following. Then we show why standard solutions are not always suitable for WSANs and in particular in the context of industrial automation. Since even two instances of wireless implementations of automation systems can differ substantially, we finally conclude that first, there is no one-fits-all solution and second, that a clearly distinguished definition of the requirements is needed in order to find a suitable system architecture.

2.1 Information Security

Information security describes the properties of information systems, which ensure confidentiality, integrity and availability. In WSAN based automatic systems is the information security particularly an economic factor. It defines the cost for an attacker to get important business information or to disturb the error-free operation of an industrial plant.

The three terms –known as CIA triad, Figure 1– confidentiality, integrity and availability are the core principles of information security. In this section we present a short description for these principles.

Confidentiality is the ability to ensure that information is accessible only to authorized people or systems to have access.

Integrity is the ability to ensure that data is an accurate and unchanged representation of the original secure information.

Availability is the ability to ensure that data are readily accessible to the authorized all times.

For an error-free operation of a facility it is very important that information are trustable and accessible all times, since measured data and controlling information regulate the workflow in sensor and actuator based industrial plants. If data regarding the facility's workflow can be easily obtained by an attacker, it would pose the feasibility to gather important business information with a minimal investment. That would imply a maximal benefit for him.

The authors of [3] described four additional key security concepts we will also follow in our evaluation, since there are as well of some importance to automation systems.

Authentication concerns the verification of the peer's identity.

Authorization checks whether the peer has permission to conduct some action.

Accountability makes sure the actions can be assigned to the corresponding communicating participants.

Non-Repudiation is the undeniably of an action.

In the automation environment it can be the question whether a received sensor reading or controlling command is actually from the right sender and not a forged message. For modern computer system the principle of least privilege more and more takes hold. The authorization is an instrument to enforce this principle. For sensor nodes in addition it can reduce data processing efforts because invalid packets do not need to processed in higher layers.

Accountability and non-repudiation are very important for e-commerce and e-business systems. In WSANs these properties can be mostly covered with the mechanism used to ensure integrity.

Table 1. Standard mechanisms for the seven protection goals

Protection Goal	Mechanism
confidentiality	encryption
availability	redundancy, filtering
authorization	passwords, filtering
authentication	signatures
integrity	secure hashes
accountability	audit, logging
Non-Repudiation	signatures, logging

For all security goals mechanisms have been developed in the past. Table1 gives a short overview of these mechanisms. For instance a standard means that is supposed to provide confidentiality is encryption. However, that such a mechanism satisfies the specific goal does not mean that it works under all circumstances. Most approaches have initial assumptions regarding the communication channel, the peers and the environment. If and how a mechanism works depends heavily on environment and application. For example it is common knowledge that cryptography provides confidentiality over an insecure communication channel, but it does not provide confidentiality for measuring on the sensor device. The problem becomes even more apparent for ubiquitous devices that theoretically can be picked up by everyone[11]. On the other side confiden-

tiality in communication networks can be provided without encryption if it is defined that no entity without authorization can read from the network.

So, each of our seven protection goals can only be achieved if the initial assumptions under which the implementation of the system are all kept over the life time of the system.

2.2 Security in WSANs

With the substitution of wired communication with wireless communication –not only in the context of automation– also a lot of new security threats emerge. First of all there is the insecure open channel. Everyone in radio transmission range can eavesdrop messages. Also everyone in range could interfere, i.e. send or modify messages. This means that protocols such as industrial Ethernet and field buses do no longer satisfy all requirements with respects to reliability and security under the new conditions. The access to critical systems cannot be controlled by administrative guide lines or physical barriers anymore.

Consequently, the embedded security means do not hold in the new environments. The first idea would be to change the existing protocols so that they fit in the new environment. But that is rarely possible for existing industry plants. Instead usually new components and sub-systems, like the wireless system, have to be adapted in order to be attached to the existing infrastructure. Therefore it is essential that the new wireless components ensure security parameters in such a way that the security and dependability requirements of the existing system will not be compromised.

In the following we introduce two examples of existing automation systems that should be extended with wireless networks. Regarding protection goal security, we will clarify that existing mechanisms of wired networks can not be transferred directly to WSAN systems. Instead it will be necessary to add additional mechanisms to reach the same level of security.

2.2.1 Example Description: Waterworks

The current architecture of the given waterworks facility consists of a wired network which connects a set of sensors on wells, filters and clear water pumps with decentral nodes, a programmable logic controller (PLC) and a central monitoring and control station. Industrial Ethernet connects PLC, the central station, and decentral peripheral nodes. The sensors are directly connected peripheral nodes.

The basic idea –shown in 2– is to replace the direct link of the peripheral nodes by a wireless connection. The sensors will be placed in a wide area with a long distance to the peripheral nodes. The links are used for measurement data and controlling commands. The captured data will be the base for the controlling commands and a smoothly operation of the system.

The system has to achieve the following basic security requirements:

- Confidentiality of captured data
- Authenticity of communication peer
- Access control for open access points

For wired infrastructures only two security problems are imminent: first the internal unintended misuse, which can be solved by training the administrators and using an easy to use and clearly user interface, and secondly a potential connection between the productive and a public accessible network. The latter can be solved by using a very strict firewall or physical isolation of productive from public networks. In contrast for WSANs the connection to public accessible networks is ubiquitous. The isolation of the networks can not longer be achieved by fences or by the enclosing of the systems. The attacker –insider or foreigner– can use a mobile device to penetrate the network and has to be only in the near of a wireless node or access point. Furthermore the existing problems will be more severe. Every misuse can deactivate the necessary protection system and make the whole plant vulnerable. Mobile devices become a gateway from the Internet to the sensor nodes.

2.2.2 Example Description: Robot Cell

The robot cell is an example of the factory industry and consists of a robot arm with changeable tools and a tool depot. Every tool has a different set of sensors and actuators, which are controlled by a central unit. The central unit is usually a PLC as part of the robot cell or is located in the near of it.

For the project demonstrator we use a wireless connection for the sensors and actuators of a tool. The motivation is that with wireless connections the replacement process can be faster and less error-prone.

The system has to achieve the following basic requirements:

- Covering of control data
- Authenticity of sensors and actuator

In a wired infrastructure the basic requirements can be covered without additional security means, since access to the communication network is strongly restricted by fences, production hall etc. For a WSAN this is not longer a valid assumption. The wireless components can also be controlled by an attacking unit outside the production hall.

2.3 Environment-Driven Constraints

Using standard solutions for the new security threats is hardly suitable for embedded WSANs. Industrial plants are using a lot of different communication protocols and hardware with an intense focus on dependability. Standard security solutions are often very expensive in computation and generate a significant protocol overhead. To fit a protection goal in embedded devices an adoption of

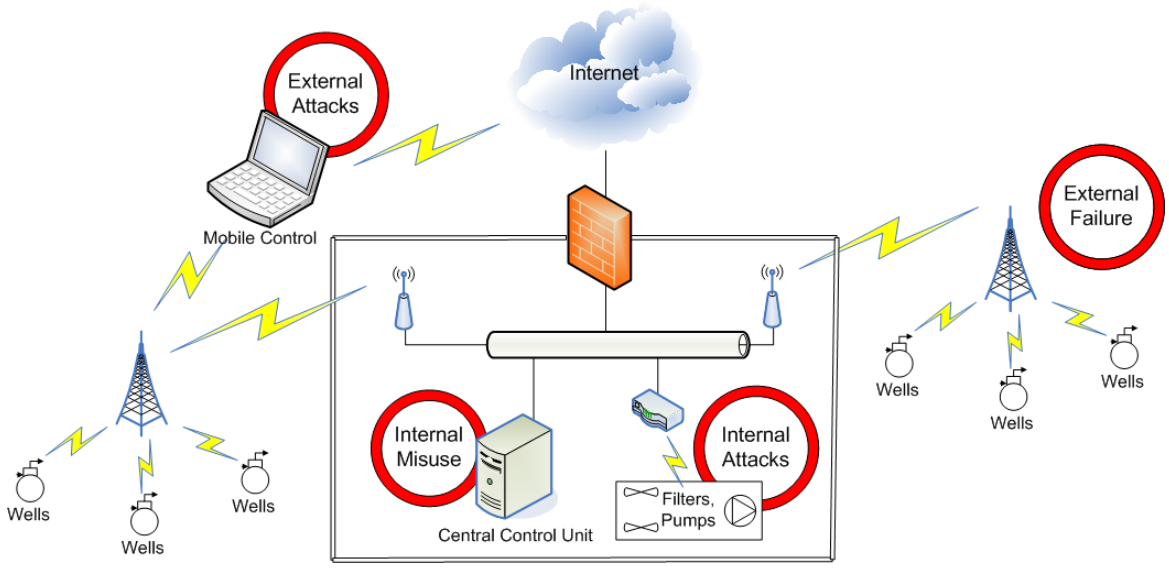

Figure 2. Wireless waterworks infrastructure

an existing solution or a distribution in hardware and software becomes necessary. Identifying the ideal trade-off is a non-trivial issue.

In both examples the new components will be connected by wireless links and hence will be vulnerable and can be miss-used as point of entrance. In this section we will explain why a good embedded security solution for the waterworks is not a good solution for the robot cell.

In both examples the final link to the sensors should be replaced by a wireless connection. The data transferred over these links are controlling and measure data. Accessing or overtaking a sensor node by an attacker can compromise the operation of both plants. Capturing data can gain a benefit in an imaginable industrial or national competition. It would be necessary to cover the security goals confidentiality, authenticity and authorization in those wireless architectures.

A potential solution for waterworks is using standard encryption for the controlling and measured data. The sensor nodes and the access points are powerful enough and have no problems with power consumption. Because of the easy accessible location of the sensor nodes especially at the wells, we need also good authentication, integrity and authorization. This can be done by signature algorithm like SHA1[4] or RipeMD and a light weight firewall [6]. To ensure the availability we can use a backup communication link, with lower performance or use another hop-by-hop route.

All these solutions do not work for the robot cell. Here we have a very small set of data in a high frequency which have to be processed by a faint sensor node. Data encryption with block size padding and additional header information will extend the packet to an inadmissible size. It would be better to use algorithm without block size binding like RC4 or a modified AES [7]. High level packet filtering in real-time will not be a feasible, the needed calculation power and caused latency are not acceptable. Authentication and authorization should be solved while registering of the new tool set. A more physical solution like a bar code would be feasible. Availability can be covered by a power down in any case of an error. A human administrator would be in the near of the system and can interfere in real time.

In this short example you can see, that the solutions for two systems with nearly the same protection goals need to be extremely different. That is mainly caused by the environmental constraints. In the field we have many more factors not described here which additionally have to be obeyed. In the next section we introduce an approach to covering this problem by a more tangible process.

3 Our Approach

Based on the perception that realizations of security-providing mechanisms usually cannot be delivered by a one-fits-all solution this section discusses a methodology that does not only respects the security requirements of the application but integrates environmental properties and safety-constraints. The result is a well-defined composition of system components that promise to satisfy the given requirements. The formalized categorization of solutions allows the establishment of a knowledge base that can be applied for the development of new systems. The results of each new engineering process will also extend the knowledge base.

3.1 Development Flow

The fundamental idea of our approach is shown as Figure 3. The result of the system analysis process is a list of *target properties* (Security goals, dependability requirements and environmental constraints). Driven by the requirements we start an iterative process that successively

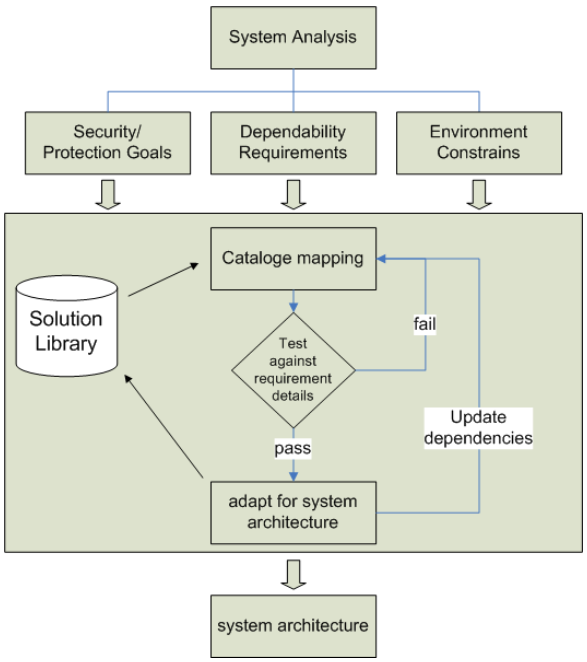

Figure 3. Flow of the selection process: Definition of Security, Dependability and Environment are input for iterative selection process. The solution library allows reuse.

takes promising solutions out of a solution library and tries to attach them to the system under development. An evaluation after each step tests the outcome of the incrementally extended system. If the extension is beneficial, i.e. the test is passed, the system architecture will be adapted. The new system –even if it does not satisfy all given requirements– will be added to the solution library, so that the knowledge base is extended for the future. After updating the dependencies and solved requirements of the new system, a new iteration of the development will start. That process will be repeated until a system architecture is found that satisfies all given requirements. This architecture will be the blueprint for the actual system integration.

3.2 Inputs of the Selection Process

As already introduced, it is the key that the goals are objective and their fulfillment can be verified.

The major problem of traditional assessment process of security requirements is an implicit fuzziness that concludes in wrong assumptions. Surveying operators of automation facilities we often got requirement statements like 'confidentiality is no problem because no one unauthorized can enter our networks'. In that case it could be a miss-interpretation of the operator to conclude that confidentiality is no issue for that facility. In fact, it can be assumed that it is an issue, but since that security related statement already includes environment and an assumption of the solution, it is not clear.

The need for confidentiality as security goal does not depend on the environment. It depends on data and a sort of degree characterizing the security strength. Consequently the requirement definition of concealment –just like for the other requirements– must be defined isolated from environmental aspects. For example the requirements regarding integrity of data in a facility are unaffected by the used network. If the facility switches from wired to wireless, the security requirements will not change, but just the environment. Indeed the eventual solution will change significantly but the inputs to our process will change just slightly.

Due to the strict separation of security, dependability and environment in our definition process we are able to pose questions that aim toward a precise and objective problem definition. At this point the questions mostly concern whether specific properties (e.g. concealment, integrity) are needed. For a precise definition process it is also imperative to define the degree of each feature. Potential metrics are the assumed cost or duration that are needed to break the mechanism. However, in order to illustrate the general idea, in this paper we stress the pure existence of specific requirement.

3.3 Mapping from Requirement to Practical Solution

The center and brain of our approach is the mapping and selection algorithm. Its task is to find combinations of system components that satisfy the requirements. Straightforward said, we want to map the three requirement descriptions into a single system.

3.3.1 The Solution Library

The solution library is a repository of potential solutions, which is used by the mapping process. The library stored entries that are very similar to classic security architectural patterns [12]. Such patterns usually describe implementation aspects of software application programming. Our patterns additionally consider environmental mapping and protocol selection so that each entry in the solution library contains the data like description of the solution including a problem statement, security and dependability implications (what do they solve), environmental constraints and parameters, dependencies, and discussions of benefits and disadvantages.

The data will provide developers the means to assemble systems out of the basic building blocks that are the entries of the solution library. Additionally the clear structure allows objective analysis of the resulting system. Although it is envisioned that the analysis can be performed in an automatic process, currently it is required that entries and structure can be handled by a human engineer. Description, motivation and discussion are part of each entry mainly for that reason. For automated integration a rather formal description of dependencies, constraint and properties would be required.

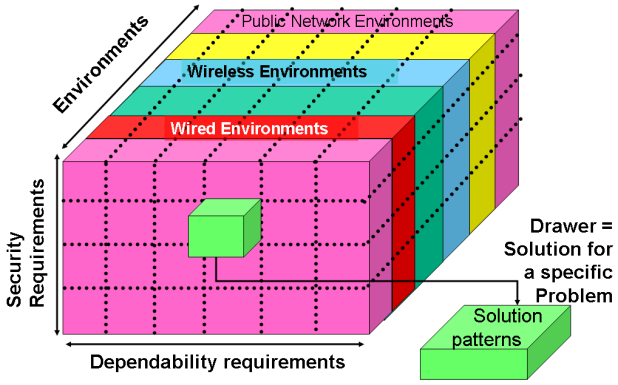

Figure 4. Three-dimensional tomography of Security/Dependability/Environmental Requirements. Each sub-square contains a set of solutions for the specified requirement combination.

3.3.2 Selection Process

Currently we consider two ways of mapping: a direct mapping, and the more generic search approach. The direct mapping from the requirements to suitable system components is illustrated as Figure 4. It shows the design space of all combinations of security, dependability and environmental (SDE) requirements as a cube. That way any combination of SDE directly points on a subcube, later on denoted as *drawer*, which contains suitable solutions that are either single patterns or a combination of patterns.

Formally the problem can be described by three orthogonal goal sets: Security (S), Dependability (D), and Environment (E). With our analysis process we determine subsets $s \subseteq S$, $d \subseteq D$, and $e \subseteq E$ that define the properties that have to be covered in our final solution. So S,D, and E are the axises of the cube and s,d, and e are points on the corresponding axis. Initially the cube is populated with the uncombined basic patterns of the solution library. We know that each single solution pattern l_x of the solution library (Set L, $l_x \in L$) points on properties (s_x, d_x, e_x); $(s_x \subseteq S, d_x \subseteq D, e_x \subseteq E)$ it fulfills. Consequently, with their properties they define the position inside the cube. Combinations of solutions will create new drawers, so that the granularity of the sub-cubes becomes more fine.

The advantage of such a tomographic approach is that it clearly shows solutions for the well defined problem descriptions. In the simplified presentation of Figure 4 another advantage becomes evident: for example changing the wired environment to a wireless one is just a shift of a layer, i.e. e changes, while d and s stay constant. For default problems –like how to achieve good concealment over a wireless channel over a specific distance– the drawers will be populated quite soon. As stated earlier, that allows the establishment of a knowledge base that can easily be re-used.

However, due to the huge space of potential SDE-combinations not every drawer will be filled with suitable solutions. Then there is the need for a more generic search as part of the mapping process. In a straightforward search we could start with a zero configuration and successively compose basic patterns until the required properties are met. Although it theoretically would lead to a satisfying system, its practicability is highly questionable. First, the design space becomes very large, so that we assume that the complexity is hardly manageable. In particular the definition of the result of the combination of security properties is still a pending question. Though some research has been done in this area [2], a fully automated assessment process is still missing. Hence the selection process is mostly a semi-automatic process: a computer looks for possible solutions, but an engineer has to check and resolve. So starting from zero can be a very time-consuming task. Anyway, the fundamental idea of engineering is to efficiently reuse existing knowledge – a principle we also pursue in our approach, instead of starting from zero all over again.

Basically, if the content of a drawer is not satisfying we start looking for solutions in the neighborhood of the drawer. We can say that solutions found there are pretty similar to the one we are looking for and could be the basis for further development.

The idea is already shown in Figure 3. First, one would look for a means out of L that could help satisfying the target properties. If the direct mapping does not provide a result directly, the mapping process will start removing or altering single target requirements (s,d,e) so that they point on neighbored cube cells. If the newly-found potential solution passes the check (i.e. it is beneficial), it will be added to the current configuration. If the new system module has dependencies or interoperability issues, they will be resolved. In practice this 'update of dependencies' means to identify the differences between the system under development and the initial target properties, and to start a new search process with the differential target properties. If it is not possible to solve any dependencies or system requirements, the algorithm will backtrack and continue looking for other solutions. In case no solutions are found in the neighborhood of the original drawer, the search space will be further extend. In worst case we start with single properties out of a zero configuration. But usually a better starting position will be provided by the concentric generic search process.

If eventually a configuration is found that satisfies the given requirement, it will be added in the corresponding drawer of our tomographic cube. Anyway, we also propose to add intermediate results to the solution library. Even if a configuration only satisfy five of seven requirements, it still can be useful for future applications. Storing intermediate results would allow to review the made decisions, and it enables re-using security-related design solution on a broader scope.

4 Applying the engineering process

In this section we study two use cases of the waterworks example and use our approach to demonstrate, how to find a feasible solution that eventually will be integrated in the infrastructure.

4.1 Basic Use Cases of Waterworks System

The waterworks process basically is the control of pumps based on measured. From that we can derive two use cases. First, gathering measure data (uplink). The waterworks components wells, filters and pumps provide measurements, which have to transfered to the central controlling unit. The data will be gathered in a polling process every n seconds. Additionally in an emergency case an alarm has to be sent immediately.

The second use case is the control of the components (downlink). The central unit sends short controlling commands to the components and expects an acknowledgment. This command has to be executed directly, i.e. a small latency is required.

4.2 System Analysis

Based on these use cases we can derive the following protection goals.

For the uplink the data will be transferred from the wells in the fields to the water work. The data has to be concealed and a secure integrity protection is required. The dependability requirements are a short startup time in an emergency case and the periodic sending of a dataset. The environment constraints are a long distance between nodes, existing protocols for the wired infrastructure, the demand on a cheap communication link and a long node life time.

For the controlling channel the security requirements are an authorized access, an authentication of the controlling unit and the integrity of the controlling commands. The safety requirement is short response time for the acknowledgment. The controlling commands are generated by a PLC, which is connected to a wired network with a standard protocol. The protocol on the PLC should not be changed.

4.3 Selection Process & Solution Library Extension

After the classification of the inputs, we can go into the selection process, as described in Section 3. As first step we use direct mapping and check the results. If it does not match we use a generic search to find a better solution.

For the uplink case, without environmental constraints the solution library would deliver a virtual private network based on public available network services, like GSM or UMTS. However that solution would not cover our cost requirements. VPN needs costly hardware and public networks need monthly fees. Considering that the drawer of our tomographic cube with the right environmental requirements is empty, we have to start the generic search process.

We start with the proposed solution and try to replace the modules responsible for the mismatch. Since the distance between sensors is not more than 50m we can look for short range protocols. This selection gives would give us for example IEEE 802.15.4. It already has a build-in security (AES) that can be used for the encryption of the data and thus would satisfy the concealment requirement.

The access point with a possible longer distance to a node can be reached by a hop-by-hop connection over the sensor nodes. The new introduced hop-by-hop communications extends the given solution in such a manner that we can add it in draw for long distance communication. The new solution requires merely a multi sensor architecture.

For the uplink case we need an authorized access, which is not fulfilled by IEEE 802.15.4. A neighbor draw contains a solution for mobile devices with IEEE 802.11 in a static network architecture. This solution uses the light-weight firewall on every node to reduce the computing time for packet processing. We decide that it is simple to adapt this solution to IEEE 802.15.4, so we can still use our previously found solution.

Finally, we can add this solution into our solution data base with the additional environment constraints static infrastructure and small sensor node. In case even less-powerful devices should be taken the presented solution would be found in the solution library and could be further refined. Anyway, the example gives an impression how a reliable and objective requirement-driven engineering process for secure systems for industrial automation systems could look like.

5 Related Work

In this section we outline results of related work that can provide ideas for further refinement of our approach. Though it is generally accepted that security issues can only be solved in holistic approaches including security protocols, physical environment and general policies, scientific solutions covering the full spectrum are rather rare. Even the question if a given system implementation satisfies the security goal in a well defined environment is still a challenging question.

In [5] the authors discuss the integration of security aspects into a formal method based development of networked embedded systems. The focus of the security analysis language (SAL) is merely on information flow between networked entities. By that it might be a way to model security requirements of applications and to verify whether or not the correct security modules were selected.

VEST [10] (Virginia Embedded Systems Toolkit) focuses on the development of effective composition, configuration, and the associated dependency analysis. The tool helps the developer select and compose software components to a product. The analysis part even allows checking security properties, though it does not provide formal proof of correctness. Rather it applies key checks

and analysis to avoid many common problems.

A tool that could be an example for a small solution library, entirely focuses on 'Security Through Usability' and has been published by the CRISIS project [1]. The authors categorized several key distribution schemes for sensor network applications. Based on the user inputs a suitable selection of protocols is presented. After entering main and secondary properties, e.g. small memory, connectivity, scalability, resilience, the tool delivers a list of key distribution schemes that fulfill the requirements. Additionally the tool lists specific advantages and disadvantages of the algorithms, so that competent users have further information supporting the selection process.

Security architectural patterns are discussed in [12] and [9]. Though the context of the studies is not as broad on system level, studying the proposed terminology can help improving the definition of our solution library. For example the notion of a security degree as part of a pattern description as described in [9] can be valuable for the objective security assessment process as it is required in our selection algorithm.

Composition of security mechanisms is discussed in [2]. They propose a framework that breaks down the security protocols in atomic cryptographic tasks that can be combined to composed protocols. An application of the idea inside the selection algorithm as well as an extension of the described cryptographic task towards combinable building blocks for safety and environment could be a promising approach.

6 Conclusions

In this paper we have presented a holistic approach for engineering security solutions for automation networks. One of the main innovations is the inclusion of non-security parameters such as *dependability* and engineering constraints resulting from existing systems, which we call *environment*. The latter is extremely important since it allows to explicitly model implicit assumptions, e.g. about confidentiality which is given if a system is physically shielded from its environment, which holds no longer true if wireless communication is used. The second innovative aspect is the semi-formal search for security solutions guided by the above mentioned constraints. We have illustrated our approach using real life examples currently under development in the RealFlex project.

In our future research work we will focus on the following issues: formal description of the system properties in all relevant categories i.e. security, dependability and environment. Then we will develop an automated testing functionality for checking system properties during the search for solutions.

Acknowledgment

The work presented in this article was supported by the German Ministry of Education and Research under grant 01BN0711D and by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^o 225186.

References

[1] C. Alcaraz, J. Lopez, and R. R. Castro. Choosing a key distribution protocol for your sensor network. http://www.lcc.uma.es/˜roman/KMSCRISIS/, Jan 2008.

[2] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *IEEE Symposium on Foundations of Computer Science*, 2001.

[3] N. Daswani, C. Kern, and A. Kesavan. *Foundations of Security: What Every Programmer Needs to Know*. Apress, Berkely, CA, USA, 2007.

[4] D. Eastlake 3rd and P. Jones. US Secure Hash Algorithm 1 (SHA1), 09 2001. RFC 3174, Informational.

[5] M. Eby, J. Werner, G. Karsai, and A. Ledeczi. Integrating security modeling into embedded system design. In *International Conference and Workshop on the Engineering of Computer Based Systems*. IEEE, 2007.

[6] P. Langendoerfer, K. Piotrowski, S. Peter, and M. Lehmann. Crosslayer firewall interaction as a means to provide effective and efficient protection at mobile devices. *Computer Communications*, 30(7), 2007.

[7] National Institute of Standards and Technology. Advanced encryption standard. *NIST FIPS PUB 197*, 2001.

[8] RealFlex Consortium. RealFlex: integration of reliable wireless communication systems within sensor/actuator networking in automation systems, http://www.realflex-projekt.de/, 2009.

[9] D. G. Rosado, E. Fernandez-Medina, M. Piattini, and C. Gutierrez. A study of security architectural patterns. *Availability, Reliability and Security, International Conference on*, 0, 2006.

[10] J. Stankovic, R. Zhu, R. Poornalingam, C. Lu, Z. Yu, M. Humphrey, and B. Ellis. Vest: An aspect-based composition tool for real-time systems. In *Proceedings of the IEEE Real-time Applications Symposium*, 2003.

[11] D. Westhoff, J. Girao, and A. Sarma. Security solutions for wireless sensor networks. *NEC Journal of Advanced Technology*, 59(2), June 2006.

[12] J. Yoder and J. Barcalow. Architectural patterns for enabling application security. In *Fourth Conf. Pattern Languages of Programming (PLoP)*, 1997.