

Functional Safety and System Security in Automation Systems – A Life Cycle Model

Thomas Novak
Vienna University of Technology,
Institute of Computer Technology
Gusshausstrasse 27-29
1040 Vienna, Austria
novakt@ict.tuwien.ac.at

Albert Treytl
Austrian Academy of Sciences
Research Unit for Integrated Sensor Systems
Viktor-Kaplan-Strasse 2
2700 Wiener Neustadt, Austria
Albert.Treytl@oeaw.ac.at

Abstract

Industrial and building automation systems are more and more important in industry and buildings. New services and novel fields of application call for dependable systems. Two very important properties of such a system are functional safety and system security. In the opposite of today's development where safety and security are treated separately, investigating security together with safety leads to a reduction of effort in the different phases of system life. That is because they have some similar objectives, but realized by different measures. The intention of the paper is to present a way of developing a safe and secure system as well as to show the associated benefit with special focus on building automation.

1. Introduction

Historically, safety and security issues have been treated separately. Especially, safety systems are setup and operated totally disjointed from other systems. That is, each domain has its own physically separated system and gateways are required to connect them. For safety systems additionally absence of reaction is required and has to be proven usually resulting in a limited read-only access to the safety system.

Upcoming trends such as remote access via the Internet, advanced control operations or cost reduction by using shared networks, not limited but obvious in the area of building automation, advise to rethink this separation and setup concepts for systems that allow common usage by safety, security, HVAC (heating, ventilation and air conditioning), and lighting and shading. Using redundancies in building automation control systems by integrating safety critical, security relevant and standard operation into a single communication system would allow for big savings.

But these trends break up the isolated structure of existing networks and therefore introduce new risks and

threats concerning safety and security and set new challenges to the safety and security measures in existing systems. Concerning security today's automation systems and networks are in general lacking efficient security features. If available security is an extensions that is seldom used and often has non-negligible drawbacks (e.g. [1],[2]). In the same way automation systems lack native support for safety and have been enhanced with safety features on application level (e.g. [3]). What is in common is that dependencies between different operation modes, and safety and security in particular are not considered. Safety and security are examined independently not considering potential hazardous side effects on each other.

In a first approach embedding security measures are to guarantee the correct execution of all safety relevant operations in a mixed operation environment.

This goal requires that the systems offer a flexible security framework that on the one hand handles the correct usage of resources needed by safety, e.g. QoS parameters like keep alive intervals, communication times, the implicit access control demanded by the safety application, and on the other hand also offers the same and other services like access rights or authentication to other applications.

This paper introduces a possible common approach investigating security together with safety throughout the whole system life that on the one hand leads to a reduction of effort in the different phases of development and on the other hand aims at the design of systems which have security and safety in their core. The intention of the paper is to present a life cycle approach based on ideas mentioned in [12] – using existing international standards, IEC 61508 [7] and IEC 15408 also known as Common Criteria (CC) [11], and their methodologies – that allows for a combined approach towards security and safety. The particular focus is set on safety and security in building automation.

The remainder of the paper is structured as follows: Section 2 describes security and safety goals with their commonalities. Section 3 introduced the safety-security life cycle model. Finally, section 4 presents rules for

conflict resolution and a measure assessment. The practical examples given to illustrate the concepts are taken from SafetyLon-Project [3] that strives for making LonWorks safe, from the authors activities related to secure industrial and building automation systems and the standardization of safe and secure systems in CEN.

2. Common Procedure: Integrity, Authentication and Authorization

Looking at the goals of safety and security similarities can be identified that show potential for a combined synergetic approach. In particular the common procedure of integrity check, authentication and authorization is a main building block for applications in both areas.

2.1. Definitions

“*Functional safety* is part of the overall safety that depends on a system or equipment operating correctly in response to its inputs” [7].

Security is concerned with the protection of assets from threats, where *threats* are categorized as the potential for abuse of protected assets” [11] In the domain of security there are two main subdomains: network security or internet security, and system security or computer security [15].

2.2. Safety and Security Goals

In order to understand the potential of synergies given between safety and security the goals of the two areas should be analyzed:

Safety goals are

1. Integrity, which demands the correct operation of the system under all defined circumstances with in a fixed period of time. It is usually divided into stochastic (hardware) integrity and systematic integrity.
2. Authentication, which demands that a message is coming from the correct source. A common approach is source based addressing.
3. Availability is not necessarily a direct safety goal since a non-available system can find a simple fail-safe state by going to no-operation, yet for practical reasons this trivial solution is not desired.
4. Authorization is usually implemented implicitly by allowing authenticated operation. Additionally, a check for maximum plausibility is sometimes applied, for example to check timing values.

Security goals are

1. Confidentiality, meaning that only authorized entities must be able to read confidential data,
2. Integrity, stating that no unauthorized entity must be able to change data without being detected,
3. Availability, mandating that data is on-hand when it is needed

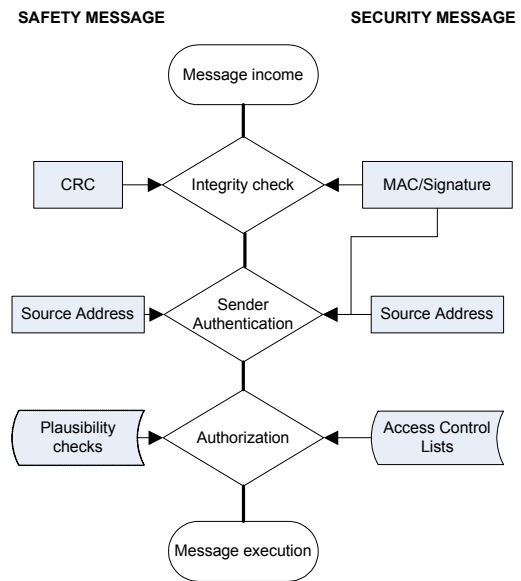


Figure 1 Common procedure: integrity, authentication and authorization

4. Authentication, allowing to determine the sender/creator of a message
5. Authorization, defining access rights.
6. Non-Repudiation, giving evidence that the sender/creator of a message issued the message.

Looking at the security goals relevant for automation systems confidentiality and non-repudiation¹ can be neglected for the systems of interest in automation ([1],[2],[4]). Hence, comparing the remaining important security goals with the safety procedure a common pattern between safety and security can be identified.

A chain starting with integrity verification, authentication followed by an authorization is given (see Figure 1). Measures in this chain are differently implemented in safety and security since they aim at different sources of failures, but pursue the same goals; Security aims at protection and defense of threats from intentional attacks, safety measures are a protection against malfunction of the system itself including fault tolerance, safety integrity and fault resistance [5].

E.g., safety authorization is based on maximum plausibility, i.e. is the value within a fixed range, or even simpler allows everything that passes the authorization stage. In contrast, security demands fixed access control lists (white or black lists). Similar for the integrity, CRC (cyclic redundancy check) codes computable by everyone are sufficient against stochastic failures, whilst security necessitates cryptography that requires the knowledge of a secret key to properly verify the integrity of a message. Looking at this example replacing the

¹ Non-repudiation is of big importance and rarely used in actual operation. It is only used as posterior measure for tracking, e. g. billing or making lists of accesses of maintenance personal.

individual measures for integrity protection (cryptographic message authentication code (MAC) and CRC) with a jointly used MAC fulfills the requirements for both domains.

Yet, finding these commonalities is not that straight forward since measures can need different efforts or even find no common equivalent. Optimization goes much beyond simple replacements; they need a holistic analysis that requires considerations of safety and security in the complete life cycle from development, to operation and disposal.

3. Life Cycle Model

A life cycle model is a structured and systematic model covering the *development* and *use* phase of a system. Within the paper a life cycle model is used as generic term for every procedure starting with a product's conception and ending with its disposal. Or, in other words a life cycle model specifies a logical activity flow of a project.

3.1. Life Cycle Approaches

A life cycle approach has become 'best practice' in the safety domain. Examples of safety life cycles can be found in [6] or in the international standard for functional safety IEC 61508 [7]. There has been a common understanding that activities such as fault avoidance and fault control must be applied at different stages of the life cycle. Often safety assessment work has been confined to assessing whether the proposed architecture meets the target failure probabilities [8]. Less attention was paid to the installation, maintenance and disposal phase [9]. Therefore, a lot of serious safety problems occurred during that phases.

Similar approaches, albeit less detailed and accepted, have been development in the security domain. In [10] it is outlined that building any type of software securely is only possible if security issues are considered during all phases of the life cycle. Hence, seven so-called touchpoints (a set of best practices) are introduced, each of them applicable to a different life cycle phase. In the international standard IEC 15408 [11], also known as Common Criteria (CC), security related activities for the various life cycle phases are specified. They are organized in classes such as activities for requirement specification or installation. Additionally, CC specifies a way of how to receive security requirements. In general, the trend to ensure security during the various stages of a product life is clearly perceptible. Its importance is growing due to the increasing complexity and connectivity of security critical systems

Whereas in the safety domain life cycle models are specified, the security domain very often determines activities relating to security, but does not define a model, i.e. the flow or order of activities.

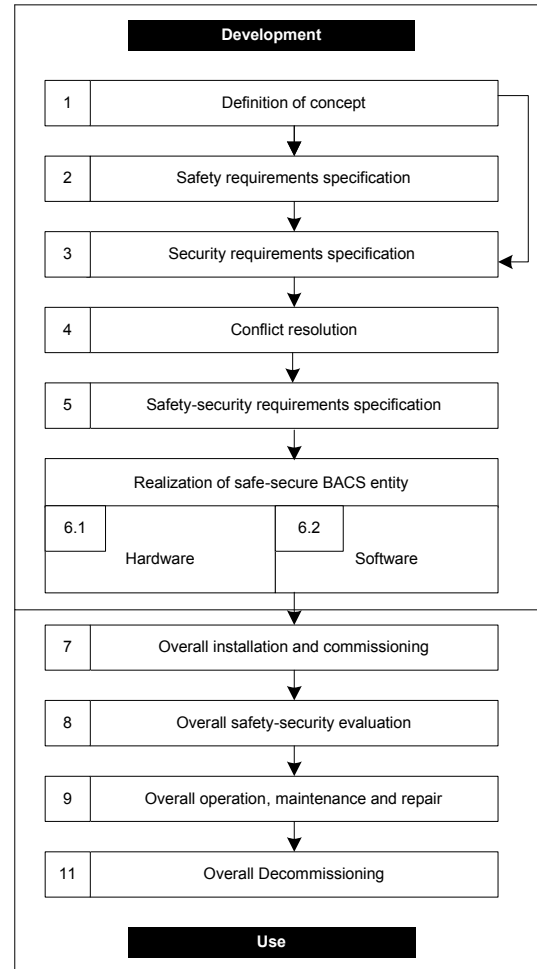


Figure 2 Safety and security life cycle model

The basic idea of the safety-security lifecycle presented in Figure 2 is to use the safety lifecycle from IEC 61508 and integrate the security approach specified in Common Criteria (CC). Requirements how to proceed are given for every phase of the system life. Moreover, activities are added to consider safety and security dependences resulting from integrating safety and security.

The two international standards IEC 61508 and CC are not application specific. They are providing a set of requirements and procedures to be used in different applications. They are chosen as basis of the proposed life cycle model because they have a good coverage of functional safety and security aspects. And the standards are considered to be well-accepted in their domains. In addition, both make a classification and a comparison of systems possible by specifying different levels: four safety integrity levels for safety related and seven evaluation assurance levels for security related systems. The rigor and amount of requirements increases with higher levels.

The safety-security life cycle model is divided into two major parts:

1. The first part includes activities related to the development of an entity in an automation system, e.g. a node. The development phase includes safety dependent and security dependent activities. In addition, an approach to deal with conflicts resulting from contradicting safety and security requirements is integrated into the development phase.
2. The second block is related to the use of the automation system and refers to all activities during the use of the complete system.

Although activities of both phases are shown in a sequential way in Figure 2, the life cycle model only intends to show that activity $n+1$ requires input of activity n . For the sake of readability recursions that will occur with the life cycle are not included in this figure and activities may be required to be redone during system life to receive a safe-secure system, e.g., conflict resolution (discussed later in the paper) may reveal a conflict that has an impact on the hardware environment. Hence, safety and security requirements have to be investigated again.

3.2. Development Phase

The development phase is the first phase of the safety-security lifecycle model. Stages 1 and 2 are following IEC 61508, stage 3 is following the Common Criteria. Stages 4 to 5 are additional new activities to handle dependencies between safety and security. These stages are of great importance since the steps defined here set the base for the second use phase. In the end, stage 6 deals with the realization of an entity.

3.2.1. Definition of the Concept:

As mentioned in [12], the development phase begins with definition of the concept that is input to the safe dependent and the security dependent part. First of all the purpose and the scope of the automation system in general and its entities in particular must be defined. It is important to know what the BACS is used for, its field of application. Additionally, it is required to specify the scope: Are there 10000 or only 100 nodes in the automation system? Are they connected to an intranet or even to the Internet?

3.2.2. Safety Requirements Specification

The safety dependent part starts with a safety scope definition where the boundaries of the entity in an automation network are determined. Next a hazard and risk analysis is performed within the aforementioned boundaries. The hazards can result from failures on the network such as delay of messages or result from failures on the entity like memory failures. The hazards are identified and the risk coming from the hazards is assessed. If the risk is not acceptable, i.e. higher than the target safety level, safety functions must be specified to reduce the risk, e.g., a CRC implemented to detect stochastic failures. Requirements on such functions are

the output of the safety dependent part comprised in the safety requirements specification.

3.2.3. Security Requirement Specification

Results from the safety investigation and definition of the concept are input to the security dependent part of the safety-security life cycle model. First, the security environment is examined: the physical environment and assets, i.e. information and resources that require protection. Typical examples of assets in automation systems are sensor or actuator data. Next, the assets are valued and threats to them are specified. A typical threat to sensor data is deliberate manipulation. Moreover, the risk resulting from a threat is estimated. The measures to reduce the risk or in other words requirements on the security functions required to protect the assets are specified according to the value of the asset and the risk of threat to the asset. E.g., to detect manipulation of sensor data a message authentication code (MAC) needs to be integrated.

3.2.4. Conflict Resolution:

Safety requirements and security requirements are investigated in the conflict resolution activity. Interaction between the different kinds of requirements is examined. Conflicts between safety and security requirements need to be resolved. Section 4 introduces a conflict resolution policy for this.

3.2.5. Realization of a Safe-Secure Entity

This activity is separated into hardware and software realization as it is common practice and suggested in IEC 61508. Very often realization means to enhance a standard automation system with safety-security functionality ([2],[3]). Hardware realization deals with the design of a hardware architecture including standard components such as a standard network access unit on a node, and the development of programmable and non-programmable hardware. For example, a node in an automation system uses a two channel architecture ([3],[13]). Safe-secure software to be integrated into the software of an existing automation system is located above layer 7 (application layer) of the ISO/OSI reference model [13]. Examples are PROFIsafe and SafetyLon, or an approach presented in [1] to secure LonWorks. Such approaches have the advantage that they do not require to change the layers of the standard protocol stack and allows for interoperability aspects.

In general, hard- and software realization applies standard mechanism in development. However, the requirements on the quality are higher compared to standard development. E.g., software artifacts have to be tested with a great amount of test cases to reach a high testing coverage.

3.3. Use Phase

The use phase is concerned with the installation, commissioning, operation, maintenance and disposal of the automation system. In opposite to the development phase which is unique for every entity of the system such as the node or gateway, in the use phase activities are relating to the whole automation system and must consider the overall interaction among all entities installed in a particular installation.

3.3.1. Overall Installation and Commissioning

In this activity the different entities are integrated into the automation system. Therefore, the safe-secure entities must be identified clearly in order to transfer the communication parameters to the designated nodes. The parameters like timing values are used to handle the safe-secure communication. Cryptographic keys applied to perform cryptographic operations like calculating a MAC must be distributed to the various entities.

3.3.2. Overall Safety-Security Validation

In the field of safety (IEC 61508) the summary of safety requirements is considered to be the specification of intended use and system requirements. Hence, safety validation is the process of comparing system behavior with the safety requirements specification(s). In the context of security (IEC 15408), security objectives are a statement of intent. Seen from a more general point of view, safety-security validation is concerned with investigating if risks have been mitigated properly and the risk mitigation strategy is working. E.g., validation means checking if integrity of the node is ensured constantly.

3.3.3. Overall Operation, Maintenance and Repair

Operation covers all activities required to operate the BACS in a safe-secure way. As a consequence, the issues like the ones mentioned next should be addressed: how to test that the data transferred during commissioning was successfully stored on the designated entity; how to switch from commissioning to operation of the system; how to update cryptographic keys; and finally how to recover from network failures due to stochastic or systematic failures, or intentional attacks.

Maintenance and repair comprises activities such as how to gather diagnostic information in order to react to failures. Another topic is the replacement of a safe-secure entity, or modification of communication parameters. Maintenance regarding reconfiguration of an automation system in general is a very challenging task. Just think of an airport with ten thousands of nodes. It is not acceptable to shut down the complete system when a node shall be replaced or configuration parameters shall be changed, just because safety and security ought not to be endangered. Sophisticated management of maintenance is a necessity.

3.3.4. Decommissioning

It is the last stage of the safety-security lifecycle. There are three ways of understanding decommissioning: The whole system, an entity or a logical communication path between two entities can be decommissioned. Similar to maintenance a ordered decommissioning needs to be performed to maintain the safety-security of the system.

An important fact of the presented parts of the use phase is that it can only facilitate measures that are planned and implemented in the development phase. If new (safety-security) requirements arise, an appropriate return to the development phase is required in order to solve the conflict.

4. Conflict Resolution Approach

Integrating safety and security causes more or less interaction in every stage of the life cycle – both coincident goals and conflicts. Identity (coincident) means that safety and security strive for the same with equal or different effort. In opposite to identity, conflict indicates that safety and security pursue contradicting things. Of course, there are also areas where they do not interact. These requirements are then considered to be independent.

Conflicts or identities between safety and security are always result of conflicting or identical requirements or conflicting measures due to identical requirements. Consequently, to figure out and resolve conflicts between safety and security requirements, it is necessary to examine interdependencies. That is why an activity for conflict resolution is integrated into the development phase (Figure 2).

After activity 3 of the development phase safety requirements were specified that are part of the security environment. Additionally, a set of security requirements is available already considering safety requirements. What is still missing at this point is a cross-checking of both sets of requirements. Are the safety and security requirements complementary? Do security requirements contradict safety requirements and vice versa? Therefore the conflict resolution approach at the requirement level is applied. The result is a conflict-free set of requirements.

Next, the measures implemented to satisfy the conflict-free requirements are checked. Only these measures are cross-checked that are different although they result from the same requirement, stated once during safety and a second time during security requirement specification (Figure 3). After measures assessment has been performed, a threat-hazard and risk analysis is carried out to verify the correctness of the decisions made during conflict resolution and measure assessment. Especially, the conflict resolution policy is checked if it delivers the appropriate result with regard to the field of application of the automation system.

4.1. Requirement Level

Even though safety and security have the same major goals as mentioned in section 2, they reduce risk to the goals because of different reasons. Put succinctly, safety is concerned with reducing risk to the system itself, whilst security strives for minimizing risk to information and resources referred to as assets resulting from malicious attacks. Accordingly, it is very likely that requirements how to minimize risk differ. Even worse, it is almost inevitable that these requirements contradict each other. Hence, a methodology has to be specified that presents a clear and concise, and easy to handle way of conflict resolution. It requires a specification of a conflict resolution first, a separation of requirements into two groups next to perform the conflict resolution itself afterwards.

4.1.1. Conflict Resolution Policy

A conflict resolution policy must be specified. It is a set of rules that enables the developer to allow for the particular point of view. The rules specify which requirement has to be preferred in a particular situation. A conflict resolution can be unique for the complete automation system, a subsystem like a node, or even for a part of an entity in the system (e.g. the firmware). Within the paper the policy consists of two rules applicable to a (sub)system:

1. Prefer safety requirements to security requirements if security has a negative impact on safety (see also section 4.1.3).
2. Otherwise, use security

Yet very simple, this policy reflects today's best practice and (legal) requirements for general (building) control applications.

4.1.2. Categorization of Requirements

The requirements are categorized into three groups: a list of detective, and a list of preventive as well as corrective requirements. Detective requirements aim at only detecting faults and attacks, e.g. 'Integrity of system data is detected'. Preventive requirements have the goal to prevent faults or attacks. In the safety domain such requirements are specified to avoid faults (fault avoidance). Finally, corrective requirements additionally specify the corresponding response to a fault or an attack. Preventive and especially corrective requirements may influence processes or systems and are therefore subject to investigation in the conflict resolution approach. Detective requirements must not influence the system and therefore require no conflict resolution.

The particular consequences of a requirement determine its characterization as can be seen from the following industrial control example. A security as well as safety requirement is to check the message regarding data integrity. If the system only provides reporting the violation to a monitoring station and do not influence the process the requirement is detective. In case the taken

measures will prevent the occurrence of the failure, e.g. corrupted messages are filtered, the requirement must be categorized preventive. Finally if the system reacts on the violation by, e.g., going to a safe state or actively correct the value the requirement is of the category correcting.

In practical applications detective requirements are a 'last resort' to detect faults or attacks that have not been or could not have been prevented by measures derived from preventive and corrective requirements.

4.1.3. Perform Conflict Resolution

Third, the conflict resolution itself is performed. Each requirement is evaluated regarding its action item. That is, the action and the reaction to a failure or attack is evaluated. The action item is checked against the other action items of corrective or preventive requirements. If there is no conflict, the requirement is considered to be a final safety-security requirement. Otherwise, the conflict resolution policy is applied and one of the requirements is discarded. In the end, a conflict-free set of requirements is setup.

Table 1 Corrective safety and security requirements

Failure/incident	Safety requirement	Security requirement
Hardware failure	Fail-safe state of node	Fail-secure state of single microcontroller
Integrity failure	Discard message	Discard message; after 5 successive incidents send message to network management device to signal attack.
Message lost	Fail-safe state of consumer	none
Disclosure of key	none	Stop communication until new key available

The following example is given to get an idea of the conflict resolution process. Table 1 shows four corrective safety and security requirements, respectively. The first failure is a 'Hardware failure'. For instance, the online volatile memory test revealed a fault in a sector of the RAM resulting in a failure of the first safety chip. From the safety point of view it is required that the safety chip switches to fail-safe state immediately. Since both chips absolutely must agree on every task that should be executed (two channel architecture), fail-safe state of one results in fail-safe state of the complete node. Security requirement says that the first chip has to go to fail-secure state and the second one takes over. The conflict is solved according to the given policy (section 4.1.1) by taking the safety and discarding the security requirement, since security has a negative impact on safety.

The second failure ‘Integrity failure’ results in similar safety and security requirements. Such a failure is detected due to CRC or MAC mismatches. Here, the safety requirement is included in the security requirement, since the MAC cryptographically protects the integrity (stronger measure) . As security does not influence safety in a negative way, the security requirement is chosen – rule two of the conflict resolution approach applies. Failure 3 and incident 4 have no cross impact on security and safety, respectively. Consequently, the safety requirement as a reaction to ‘Message lost’ and the security requirement to ‘Disclosure of key’ is determined to be a final safety-security requirement.

4.2. Function Level

Conflict resolution at the function level or also called measure assessment is the second part of the conflict resolution approach. Measures are derived from the conflict-free set of safety-security requirements resulting from the first part of the conflict resolution. One of the many motivations to design a common approach is to benefit from synergies on applying measures. Consequently, safety and security measures and synergies gained are classified in three groups: (1) such that directly match (i.e. safety and security measure are equal), (2) such that are unique for safety and security, (3) and such that require different efforts [12].

Measure assessment has to be only performed when safety and security measures are classified to be of group 3: different measures with different effort are derived from the same requirement. Just that group of measures exhibit conflicts regarding e.g., computational power, throughput, computation time, memory resources or application constraints. Measures of the other groups are either identical or unique and thus cannot cause conflicts per definition. The conflict resolution on function level uses six factors as shown in Figure 4 to assess the safety and security measure. In the following an example of a measure assessment is given.

In a safe system the requirement ‘Ensure integrity of data on a node or data in a message’ is granted by means of a CRC. In a secure system similar measures are used. Instead of the CRC a cryptographic message authentication code (MAC) is used that cannot be

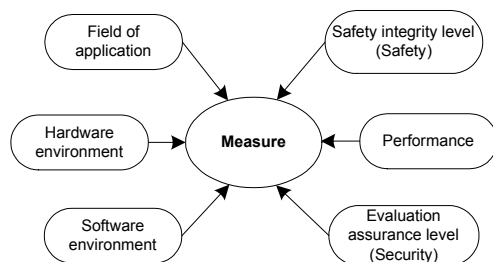


Figure 3 Measure assessment

recalculated without the knowledge of the appropriate key. According to the node address and only in case of a verified message CRC or MAC, data can be read or written.

To gain a synergy, a proper solution is to replace the CRC by the MAC since the security measure also withstands safety attacks to the integrity of data. If such a replacement can take place, it is evaluated by the measure assessment. First, safety integrity must not be jeopardized, i.e. the same safety integrity level (SIL) must be reached as it was before the replacement of the safety measure. In other words, a MAC must be selected that grants the same level of integrity than the non-secure CRC does. Second, according to the evaluation assurance level (EAL) a minimum strength of function is specified in order that the MAC chosen cannot be defeated. Third, software and hardware environment has to be considered. Using a cryptographic algorithm being implemented in hardware, results in less computational time than calculating the same algorithm in software. Furthermore, memory resources of embedded devices are low compared to PCs and have also to be considered. Finally, performance of measures and the impact on field of application are examined: Generating and verifying a 8 byte MAC on a smartcard takes about 50 ms of time whereas the process of calculating a CRC lasts about 300-500 μ s. In some applications such as fire alarm systems the difference in execution time is acceptable, but for applications like emergency push buttons with an required overall reaction time, i.e. time from pressing the button on Node A and stopping a machine connected to Node B, of less than 150 ms [13] 100ms (50ms for message protection and 50ms for message verification) is a none acceptable delay. According to the conflict resolution policy a final decision about the usage of the measure is taken.

5. Application Synergies

A main application focus at the moment is the combination of fire alarm systems and general building control. In particular application synergies could be achieved by a combination of fire system ventilation and HVAC. On the one hand fire dampers and ventilation flaps could be integrated and on the other hand the air condition can be used to exhaust smoke. A main requirement is that the HVAC must be turned off as soon as smoke is detected to prevent spreading of smoke and that the safety system gains full control of the HVAC system. Security is an enabler for the combined approach since it allows to distinguish between nodes of the safe and the secure system. Proper message authentication and integrity protection is required. For the combination fire system and HVAC the conflict resolution is rather simple since on the requirement level the safety functionality is always dominant compared to the comfort functionality of HVAC and on the measure

assessment also timing constraints in both applications are also very relaxed.

But synergies can go further: in restricted areas the access control service can be integrated into the system also setting security demands. E.g., the fire system must be able to open doors only in escape direction and the security measures must be increased in strength to avoid attacks. Also dual use of systems that detect the presence of persons are thinkable, but it is questionable if they fulfill the requirements for human safety not to lock a person in case of fire.

6. Conclusion

The possibilities to gain synergies by taking a common approach towards safety and security in building automation systems is given in many areas: at the non-functional measure level like verification and validation tools, risk analysis techniques, or testing techniques. However, it is not well-known that synergies can be gained at the requirement level or at the functional measure level (e.g. CRC vs. MAC) as presented in the paper. Gaining synergies is possible since safety and security strive for some identical goals. This fact is often honored, yet little effort to combine the fields is given because applications are thought to be either safety or security critical.

As a consequence, the paper presents a life cycle model trying to integrate safety and security in an automation system. In particular, a strategy is given how to resolve conflicts between safety and security. To address conflicts due to requirements and different implementations of measures a two step conflict resolution framework is presented, resolving conflicts at the requirement and at the function level. Such a life cycle model and its included conflict resolution are the basis of combining formerly separated networks for safety, e.g., fire alarm system, and for security, e.g., access control, and operation, e.g. heating, ventilation and air condition, or lighting and shading.

Some ideas presented in the paper have already been submitted to standardization as a working draft. Since 2007 the topic is treated as a working item in European Standardization CEN, Technical Committee (TC) 247, Working Group (WG) 4, called "Building Automation, Controls and Building Management". The goal is to create an European and later on international standard for functional safety and system security in building automation and control systems. The standard ought to be application independent and a generic standard for safety and security in building automation systems.

References

[1] C. Schwaiger, A. Treytl. Smart Card Based Security for Fieldbus Systems. In *Proceedings of 9th IEEE Conference*

- on Emerging Technologies and Factory Automation*, Volume 1, pp. 398-406, 2003.
- [2] A. Treytl, T. Sauter, C. Schwaiger. Security Measures in Automation Systems - a Practice-Oriented Approach. In *Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation*, Volume 2, pp. 847-855, 2005
- [3] T. Novak, T. Tamandl. Architecture of a Safe Node for a Fieldbus system. In *Proceedings of the 5th IEEE International Conference on Industrial Informatics*, Vol. 1, pp. 101-106, 2007.
- [4] M. Naedele. Standardizing industrial IT security - a first look at the IEC approach. In *Proceedings of Emerging Technologies and Factory Automation ETFA 2005*, Volume 2, pp. 19-22, 2005.
- [5] U. Baumgarten, C. Eckert. Mobil und trotzdem sicher?. *it+ti 5/2001*, pp. 254-263, Oldenbourg Verlag, 2001.
- [6] W.F. Bates. Safety-related system design in power system control and management. In *Proceedings of the 4th International Conference on Power System Control and Management*, pp. 15-20, 1996.
- [7] International Electrotechnical Commission. *IEC 61508 – Functional safety of electric/electronic/programmable electronic safety-related systems*. IEC, 1998.
- [8] D. J. Smith, K. G. L. Simpson. *Functional Safety – A straightforward guide to applying IEC 61508 and related standards*, Elsevier Butterworth-Heinemann, Oxford, 2nd edition, 2004.
- [9] P. Wratil, M. Kieviet. *Sicherheitstechnik für Komponenten und Systeme*. Hüthig Verlag, Heidelberg, 2007.
- [10] G. McCraw. *Software Security – Building Security In*. Addison-Wesley, Boston, 2006.
- [11] International Electrotechnical Commission. *IEC 15408 – Information technology – Security technique – Evaluation criteria for IT security*. IEC, 2nd edition, 2005.
- [12] T. Novak, A. Treytl, P. Palensky. Common Approach to Functional Safety and System Security in Building Automation and Control Systems. In *Proceedings of the 12th IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1141-1148, 2007.
- [13] D. Reinert, M. Schaefer (Publisher). *Sichere Bussysteme in der Automation*. Hüthig Verlag, Heidelberg, 2001.
- [14] A. Burns, J. McDermid, J. Dobson. On the meaning of Safety and Security. *The Computer Journal*, Vol. 35, No. 1, pp. 3-15, 1992.
- [15] W. Stallings. *Cryptography and Network Security*. Prentice Hall, 2003.
- [16] B. Schneier. *Secrets and Lies – Digital Security in a Networked World*. John Wiley & Sons, Inc., New York, 2000.