Security Challenges in Cyber-Physical Production Systems

Chapter · January 2018

DOI: 10.1007/978-3-319-71440-0_1

2 authors:

Peter Kieseberg
Fachhochschule Sankt Pölten
80 PUBLICATIONS **684** CITATIONS
SEE PROFILE

Edgar Weippl
SBA Research
328 PUBLICATIONS **3,274** CITATIONS
SEE PROFILE

Security Challenges in Cyber-Phyiscal Production Systems

Peter Kieseberg[1] and Edgar Weippl[2]

[1] SBA Research, Vienna, Austria
pkieseberg@sba-research.org,
[2] TU Wien, Vienna, Austria
edgar.weippl@tuwien.ac.at

Abstract. Within the last decade, Security became a major focus in the traditional IT-Industry, mainly through the interconnection of systems and especially through the connection to the Internet, especially for reasons of introducing new services and products. This opened up a huge new attack surface, which resulted in major takedowns of legitimate services and new forms of crime and destruction. This led to the development of a multitude of new defense mechanisms and strategies, as well as the establishing of Security procedures on both, organizational and technical level. Production Systems have mostly remained in isolation during these past years, with security typically focused on the perimeter. Now, with the introduction of new paradigms like Industry 4.0, this isolation is questioned heavily with Physical Production Systems now connected to an IT-world resulting in cyber-physical systems sharing the attack surface of traditional web based interfaces while featuring completely different goals, parameters like lifetime and safety, as well as construction. In this work, we present an outline on the major security challenges faced by cyber-physical production systems. While many of these challenges harken back to issues also present in traditional web based IT, we will thoroughly analyze the differences. Still, many new attack vectors appeared in the past, either in practical attacks like Stuxnet, or in theoretical work. These attack vectors use specific features or design elements of cyber-physical systems to their advantage and are unparalleled in traditional IT. Furthermore, many mitigation strategies prevalent in traditional IT systems are not applicable in the industrial world, e.g. patching, thus, rendering traditional strategies in IT-Security unfeasible. A through discussion of the major challenges in CPPS-Security is thus required in order to focus research on the most important targets.

Keywords: Cyber-Physical Systems, CPS, Cyber-Physical Production Systems, CPPS, Industry 4.0, Advanced Manufactoring, Security

1 Introduction

With the continuing digitalization of our economy, even the more traditional branches of the producing industry have become increasingly connected to networks in general, and specifically the Internet. While this brings a lot of new

changes with respect to new products and services, it also becomes an increasing challenge for the security of said systems. In the past, several spectacular attacks were launched against industrial environments, most notably the STUXNET [14] malware that aimed and succeeded in infiltrating a specially sealed environment and carried out an attack that was specifically designed to cause severe damage in an unobtrusive manner. Still, while STUXNET [14] might be the most famous example, typical production systems contain an increasing amount of networking infrastructure, thus becoming cyber-physical systems. This especially means that the systems are not sealed off by a so-called air-gap anymore, but have digital interfaces to the outside world, sometimes even at component level. Thus, also the attack landscape changed drastically from focusing on getting physical access to a plant towards fully digital attacks carried out through insecure entry points, which are often not even known to the factory operator. In addition, this increasing attack surface is not only a purely financial problem for the operator, but can even have a massive effect on national security in case infrastructure critical for the nation (e.g. power plants, grid components) are attacked. The recent attacks on the Ukrainian power grid may serve as a perfect example [15]. Thus, in this paper we will discuss the implications of introducing networking equipment into industrial environments from a security perspective. One major focus will be the secure introduction of networks into CPS, while the other major focus lies in protecting the production system engineering process, a topic that has not been in the focus of research until now and is typically overlooked. While this paper cannot give solutions to most of these problems, we will discuss potential solution strategies that can be the starting point for further research.

The rest of this paper is structured as follows: In Section 2 we discuss security issues in industrial environments, including a separation from IoT-systems, which are often mingled. Section 3 deals with the issue of security in the Production System Engineering process (PSE process), concerning both, the introduction of security as a step inside the PSE, as well as securing the PSE itself. The paper is summarized and concluded in Section 4.

2 Security Challenges in industrial environments

In this section we will discuss major security challenges in cyber-physical systems, especially with respect to their counterparts in traditional web based system.

2.1 Industrial environments versus IoT

When challenging the issue of security in industrial systems, the discussion is often mixed with the challenge of securing products and services based on the "Internet of Things" (IoT), especially when dealing with the Industry 4.0 paradigm. Of course there are certain similarities between IoT and Industry 4.0 installations, namely that they both often rely on the availability of cheap

electronics that can be used as sensors, actuators, control mechanisms or other cyber-physical elements. This similarity is often extended to the whole field of introducing security to industrial systems, still, in our opinion there are major differences between cyber-physical production environments and typical IoT-installations that reflect on the security measures that can be enforced.

One major difference lies in the typical life span of either of the two installations. While industrial environments are designed for long life spans typically ranging around 40 to 50 years, IoT installations are seen as more of an end user focused thing with typical consumer life spans of some few years. Of course this is not always the case as e.g. in building automation. Still, this of course does not mean that a production system is never changed during its life span and it must also be taken into account that many parts will be exchanged for spare parts during the lifetime of such an environment, still, many parameters and major parts will still be in place after decades, leading to the problem of heaving to deal with insecure legacy systems, where sometimes even the producer might long have been gone. This issue also leads to the problem of heterogeneity in industrial environments, as decades of development and technical progress might lie between different components inside the same factory, an issue that is currently not typical for IoT-systems.

Another major difference lies in the pricing of the components, where current IoT environments focus on low-cost sensors and modules that can be applied in large quantities, whereas industrial modules are often expensive. This also disallows the exchange of insecure components that cannot be fixed or isolated in industrial environments, when compared to the typical IoT installation.

Certification is another issue that is very typical for industrial environments, especially regarding the issue of safety, especially when human lives are concerned, which is rather typical for environments like steel mills and similar installations. Most certifications require the owner of the factory to undergo recertification for every major change, and often even in case of minor changes, applied to the production system. This is not only pertaining to changes on the hardware side, but also in case of software changes, making patching, which is a typical security strategy in the IoT world, very expensive and time-consuming. Thus, while IoT-systems are more similar to software-only systems with respect to patching, industrial environments typically do not have this option. This is additionally reinforced by the issue stated above on legacy systems that often either lack the performance for introducing security measures, or where simply even the knowledge on their internal system workings is lost throughout the decades.

2.2 Challenges regarding software development and networking

In the modern world of traditional IT and network security, a multitude of different mechanisms and strategies has been devised in order to provide security, as

well as to protect vital information. As outlined in Section 2.1, several standard security strategies from the traditional software world cannot be applied that easily in cyber-physical production environments.

One major difference to traditional IT systems, and one that is especially critical for cyber-physical systems integrated with the Internet or other accessible network interfaces, is the topic of patching. During the past year, the notion of *Secure Software Development Lifecycles* (SSDLCs) [11] has become a de-facto standard for the development of software, especially in critical environments, but also in many standard applications [17]. One major aspect of these SSDLCs is the notion of patching, i.e. finding and fixing bugs in software that is already shipped to customers and applying the respective changes in the form of software updates. The speed and quality of these fixes is a well-regarded measure for the overall security policy of the companies behind these products and the removal of products from the SSDLC is typically regarded as discontinuation of said product (e.g. Windows XP). The issues with patching in industrial environments are manifold and ask for completely new strategies:

- Due to safety requirements, quick changes to parts of an industrial environments are typically not allowed, nether on the hardware, nor on he software side. Depending on the regulations of the respective industry, re-certification might become mandatory even in case of very small changes in the software, thus introducing non-negligible costs and, even more problematic in the case of patches, a considerable delay in the application of the patch.
- Due to the considerable life spans of industrial systems, a lot of legacy systems can typically be found in older plants, for which no patches are delivered by the vendors anymore, or even the vendors might not exist.
- Especially in case of legacy systems with their outdated technology, the performance of these systems is often not strong enough to cater for the requirements of modern security measures like string end-to-end encryption of system messaged or digital signatures, as these operations are quite expensive in nature.
- Industrial systems are very complex environments with a multitude of modules interacting with each other. Furthermore, many factories are one of a kind, i.e. the selection of components, their interactions, specifications, but even whole subsystems are unique. Therefore, patching a module inside such a complex system can easily result in side effects that cannot be foreseen when only considering the isolated module. Testing changes for such side effect is practically impossible, changes to the system thus potentially dangerous.
- Another problem is the delivery of the patches: While modern system often rely on connection to the internet, this is often not true for industrial plants. Furthermore, many industrial production units do not allow for downtime, which is a major issue with patching, especially when the component in question is not made for being patched at all, i.e. techniques like hot-patching [19] cannot be used.

Thus, one of the major challenges in the area of IT-Security for cyber-physical production systems is the question, how to deal with prominent insecure modules inside the factory. Furthermore, many protocols still in use do not cater for security needs at all. This is often due to the fact that at the time of construction of the plant, security has not been an issue. Introducing security to this system is often problematic due to time constraints, e.g. in real-time systems, where the overhead introduced by security measures like encrypted data transfer cannot be tolerated. In addition, changes to the communication protocols introduce a lot of additional changes in all the modules that communicate with the module in question, leading to a cascade of additional changes to the system.

Traditionally this issue was solved by providing an so-called *air-gap* between the different systems, i.e. the internal systems were not connected to the Internet at all. Still, there are some problems with this strategy:

1. With the introduction of the industry 4.0 paradigm, an increasing number of modules require connection with each other or even (indirectly) with an outside network.
2. Through the exchange of parts during the regular maintenance processes, new interfaces can be introduced to modules. This is due to the fact that standard chip sets are typically used in order to lower costs. These standard chip sets typically possess a minimum set of capabilities used by many customers, in order to sell high volumes. Thus, a new spare part that was built to the same requirements than the original one might, unbeknownst to the maintenance crew, provide an additional networking interface that can be used by attackers, thus extending the attack surface.
3. Even in case of providing an air-gap, there have been successful attacks as outlined by the Stuxnet attack, but also other instances [4].

Again, this challenge must be met with additional research on how to deal with systems that were designed to be completely unreachable by external attackers and therefore cannot provide the security measures required for modern interconnected environments. It must be noted that many of these problems are especially prominent in brown-field scenarios, where existing factories with a lot of existing legacy systems are upgraded, while some of these problems might be mitigated in pure green-field scenarios by proactively selecting subsystems that already take care about security issues.

2.3 Attack detection and Security Models

The detection of attacks is one of the major issues when providing security to systems. In traditional IT systems, Intrusion Detection Systems (IDS) [2] together with anti-virus software are typical measures for attack detection, as well as information gathering on detected attacks. Many forms of intrusion detection have been devised in the past, ranging from pure network profiling to anomaly-based self-learning systems based on modern machine learning algorithms that work

by learning "typical behaviour" based on normal usage and then try to detect uncommon traffic that might hint at an attack [6]. While IDSs are a typical measure in modern IT-systems, they can cause problems in industrial environments, especially with the introduction of overhead. This is, again, especially problematic in the case of real-time systems, where even minimal additional overhead is non-negligible and, especially in combination with legacy systems, might lead to damages to the system or even introduce safety issues. The same holds true for scanners scanning for malware on the network or on the machines.

In order to be able to supply a comprehensive analysis of the whole reach of different attacks, typical systems need to be analyzed and generalized into a set of abstract models that can be used for the further analysis. This also includes the modeling of the inherent attributes, especially target controls, side parameters and requirements, in order to provide as generalized models as possible. To the best of our knowledge, there exist no generalized models related to the security of industrial systems to this date.

In order to be able to assess the security of a system based on a model, it is not only required to model the system itself, but also the possible attackers and the attack vectors, as well as the targets of attackers. Especially the latter can be very different from normal IT-systems, as industrial systems are typically far more vulnerable with respect to timing and small delays or small changes in parameters (e.g., temperature control in a steel mill) can cause serious and sometime unrepairable damages. With this differences in targets, also the attacker models will differ strongly compared to the typical attacker models of standard IT-security, which requires further investigation. Furthermore, as industrial systems often possess completely different architectures when compared to normal IT-systems, also the attack vectors require new models.

In addition, the cyber-physical world offers other classes of attacks: Due to the complexity of the cyberphysical world and especially the criticality of operations with respect to timing and other parameters, many new attack scenarios can be devised with major differences to typical goals in IT-Security. For example, typically sensors send information on the state of a physical process to a control unit that regulates the process by steering actuators, thus resulting in complex control loops. Most prominent example is the introduction of manipulated sensor information into feedback loops, as done by the Stuxnet [14]. The main issue, why this was such a successful attack, lies in the complexity of the underlying physical process, which is only understood by a small selection of experts. Furthermore, results are not binary in the physical world: The machine does not "work or not work", it might still work in case of a successful attack, but not as optimal, outside of specifications or with higher deterioration. One issue that has been previously identified as important issue is the topic of detection of manipulated sensors. This new class of attacks, which can be considered to be the "attacks of the future" with respect to industrial systems, need to be investigated before

they can be modeled in the testing process, since they are relatively new and, to this date, under-researched. Based on these theoretical foundations, models need to be devised that can be incorporated into the automated testing process.

2.4 Securing data

Often overlooked, even from an expert perspective, the protection of data is a most critical aspect in industrial systems. This does not only adhere to the classical topic of protecting sensitive personal information in medical industrial environments, but also includes data that is often overlooked: Control information for the industrial process itself, as well as sensor information. Both data streams are vital for the industrial process, not only for the flawless operation, but the optimal setting of parameters is often a critical company secret that allows a company to e.g. produce cheaper than its competition. While the control units might be the same for all competing companies on a specific market, the details on the settings often constitute a well-protected company secret. With the introduction of networking to these control units, this secret information requires special protection, as it is a valuable target for industrial espionage. Furthermore, an attacker could also gain vital knowledge on the system by monitoring sensor data, as well as even carry out specific attacks by manipulating older sensor data stored. This data assets need to be modeled in the automated testing process. Furthermore, another attack vector that is often overlooked is the testing process itself. When testing systems, they typically need to set all operational parameters in order for realistic testing, which could reveal a lot of industrial knowledge to the tester. Currently, there are no approaches tackling this problem [7].

Currently, data in industrial environments is not specially secured, as the main paradigm that was in force when designing the foundations of such systems was that the system (and thus the data) is strictly separated from any outside network, thus making special data protection unnecessary. With the advent of the Industrie 4.0 paradigm, this construction paradigm has changed drastically, still, this has not been reflected much in current designs of data stores [10]. General concepts on how to securely store information for industrial processes, especially considering control and steering parameters which are often neglected by IT-analysts in their risk assessment, thus remains an open research question, including designs for secure data stores that are specifically tailored to the needs and requirements of industrial systems.

Another research aspect related to data protection that requires further academic research is the topic of provable deletion. The topic of provable deletion of data, i.e., forms of deletion that do not allow data restore, has been neglected in Security research for the longest time. Just recently, this topic gained some attention in research (as well as in the industry) due to the General Data Protection Regulation (GDPR) [8] that requires operators to provable delete sensitive private information on request by the owner of said information (typically the

data subject). While this feature is already explicitly required by the GDPR, the regulation does give no further details on this issue, which leaves a huge gap for research: Neither does it state, what provable deletion actually is, nor does it give guidelines on how to achieve a process compliant with the GDPR. Regarding the definition, this is especially a problem for complex systems that store data on many levels and aggregation forms, including backups for disaster recovery. Especially when using more complex mechanisms like databases, which typically need to comply with concepts of data redundancy, crash recovery and so forth, no applicable deletion process exists to this date that works against all known forms of digital forensics, short of physical destruction of the equipment of course [9]. Since testing is often done using real-world data and control information in order to simulate the real-world environment, the test bed amasses sensitive data, which is required to be deleted after the end of the test procedure, or sometimes even after each test run. Thus, it is important to research methods for effective data deletion in complex systems in order to enhance the resilience of the test bed with respect to data leakage.

3 Security Challenges for the PSE process

One issue that until now has not received the respect an attention it deserves lies within the way plants are designed and constructed, the Production System Engineering (PSE) process. This process lies at the very heart of the development of new production lines (green field), as well as in the processes regarding upgrading and re-designing existing plants (brown field). Especially the latter one becomes increasingly important in an economy that is highly driven by savings and cost-efficiency, as well as through the introduction of new products and services based on the utilization of (sensor) data and other information. In this section we will discuss two major issues, (i) the introduction of security as field of expertise into the PSE process and (ii) the protection of the PSE process itself against manipulation and industrial espionage.

3.1 Current approaches in production system engineering

Planning industrial environments is a very complicated task and requires the incorporation of experts from different fields like mechanical and electrical engineering. Typically, the processes are planned using a waterfall model as described in standard literature, such as the VDI guidelines 2221 [21] and VDI 2206 [20], is used in practice for planning. Contrary to this, the applied process is often quite different from this strict waterfall model, as changes in subsequent steps need experts involved at an earlier stage to re-iterate their work. In addition to the problem of resulting mismatches between plans and actions, the whole environments has changed considerably in the past years, resulting in increasing parallelization of engineering activities [3] and increased number of engineering cycles required in order to arrive with a production system model applicable to

production system installation [12].

Furthermore, production system engineering tends to solve discipline-specific steps in a predefined (but project-dependent) sequence [16, **?**], during which the subsequent discipline bases their engineering decisions on the results of the previous steps. Usually, there is no review of previous design decisions. Thereby, it may happen that prior engineering decisions limit the decision options of later development phases. This is especially challenging for the integration of data security considerations as an additional engineering discipline [22], since decisions taken during the production process design, mechanical engineering, and electrical engineering may limit applicable security measures in a way that does not allow to fulfill security requirements at all.

3.2 Introducing Security to the PSE

Currently, the development of industrial production systems follows a waterfall model with feedback mechanisms that require a significant manual interaction and lack clear documentation. While this can be a problem for the design process as a whole, it is a fundamental problem for defining a secure environment or putting the required security measures into place, as even very small changes on the functional level in any design step can have a huge impact on the overall system. For example, an engineer might exchange a part for a functionally equivalent one, which just happens to have another chip set integrated that additionally provides a wireless LAN interface while having no impact on any other step. Thus, also last-minute changes on-site during the deployment phase must be tracked and incorporated in the security analysis.

One issue with the current model is the introduction of software at the end of the design chain after all other steps have already been finished, thus not allowing the introduction of hardware-based measures (e.g., de-coupling and unlinking of networks, hardware security appliances) without reverting to a previous step. Also, introducing security at this late step means that many previous design decisions have been made without the notion of security in mind. However, security also cannot be introduced in one of the earlier steps alone, as they are lacking the information of the latter design phases, as outlined above. The issue is that every step introduces changes to the overall system that directly reflect on security, therefore, security needs to be considered in each step, including feedback loops to all other steps in order to solve newly arising risks at the most suitable stage. How to introduce such cross-domain solutions into PSE is currently an open research question, next to how to make this process that spans several very different domains, ranging from machine engineering over electronics to IT an agile one.

3.3 Securing the PSE process

While Section 3.2 discussed the challenges of introducing security as a field of expertise into the design process of CPPSs, in this section we will briefly outline the issue of securing the PSE process itself.

The main issue is that, in contrast to the classical example of medical research in hospitals, many experts from several different domains, each with its own set of tools, taxonomies and design obstacles, need to work together, while every change in one of the domains potentially results in changes in the other domains. While this effect is also present in current environments, the problem is gaining another level of complexity when introducing the agile PSE process required for tackling security issues (see Section 3.2), as the number of changes going back and forth will increase drastically. The problem is becoming even more complex since, as outlined before, especially in security, seemingly small functional changes can result in the emerging of serious new attack vectors and/or vulnerabilities.

Furthermore, as this issue can boil down to a question of accountability, it must be ensured that the mechanisms provide actual prove as to who is the person responsible. Thus, the open research questions with respect to providing integrity focus on developing mechanisms that provide a provable tracking of changes in the design process. As an additional requirement, many of the experts in such a process have equal rights and, due to the high level of specialization in each domain, there is no central authority available to decide on the different changes. The actual state of the PSE process must be determined in a distributed manner, i.e., all experts have to agree on a specific state as the valid one. On possible solution could lie in the adoption of smart contracts and other blockchain-based approaches, as they provide exactly this functionality in other domains [5]. Furthermore, in order to seal certain parts of the PSE, some form of rights management and access control must be devised that can be applied to a highly dynamic set of assets and users, without requiring a central steering instance. None of the involved parties can assess how and to what extend the knowledge of other domains needs to be protected, i.e., each expert has to have full control over what to share and how to guard the assets. This is an open research issue in the scientific community, but possibly this could be combined with the blockchain based approaches in order to generate smart contracts that provide access control and rights management.

With respect to securing interfaces, the main differences between standard IT systems and the systems prevalent in industrial environment come into play: While standard IT systems change quickly, industrial systems are in use for decades. Also, it is not possible to make significant changes to industrial systems once they are deployed, which also holds true for a lot of industry-specific planning and design software used in PSE as well. Thus, a secure agile PSE process requires its tools to be shielded from each other, as a weakness in one of them

could impact the others, e.g., by extracting knowledge from the PSE process or introducing changes invisible to other parties. While the latter is addressed by new technologies for tracking changes, it is clear that the agile PSE process must itself be secured against weaknesses, opening up a selection of specific research questions that need to be tackled.

3.4 Securing PSE information

Industrial espionage and the loss of knowledge to competitors is one of the major obstacles in cooperative design projects. This problem is increasing further due to the ongoing specialization w.r.t. the integration of IT technologies into classical systems. Consequently, more and more experts have to get involved in the design process. This development will continue when implementing agile PSE processes, including cross-domain topics like security which extend the amount of interactions while reducing the ability to put functionalities into black boxes (i.e., hiding details). With respect to security, a lot of this hidden functionality must be disclosed in order to analyze and model the attack surface as well as possible attack vectors. Furthermore, certain steps might be outsourced to other partners, either due to cost issues or for political reasons (e.g., involvement of local subcontractors required). If numerous different partners that probably are competitors in other tenders are working on the same project, the protection of know-how is of the utmost importance.

With respect to the protection of knowledge, major research issues which to date are not satisfactorily solved are to be found in automating the measuring of protection requirements w.r.t. derived information, e.g., in the course of aggregations and other transformations. Furthermore, the question of how to abstract planning information in order to provide knowledge protection is unanswered.

One approach to knowledge protection in PSE is the generation of sealed development environments. While such approaches exist in other domains, mainly in the medical sector, the requirements and side parameters regarding such an environment are very different for the PSE process, mainly due to the issue that there is no dedicated data owner facing a selection of data analyses. The agile PSE process requires an environment consisting of many different experts that provide knowledge either directly or indirectly by applying changes and making decisions. This also means that the knowledge to be protected is far more abstract than, e.g., medical data records. Currently it is unclear how to model and quantify this knowledge, therefore we plan to do extensive research on this issue. Furthermore, sealed environments need to provide modular feature sets, i.e., a wide selection of programs of which many are outdated, unsupported, or non-standardized and user knowledge management. Channeling this complexity into a sealed environment while allowing for updates and new features to be introduced is very complicated and a challenging research task.

In addition, there is a certain amount of data that requires strict protection, ranging from actual sensor information akin to data found in traditional medical environments to meta-information and algorithms to specific configuration information that possesses great value for the owner. In order to protect these types of information, research in the area of data leak protection is required. In contrast to other data-preserving technologies, data leak protection aims at provably finding data leaks, i.e., users having access to data and distributing it illegally. Two basic concepts have been proven to be especially promising, which are (i) the use of obfuscation/diversification and (ii) the concept of watermarking.

The basic idea behind obfuscation lies in making the information unintelligible for any attacker, thus removing its value. While typically used to protect code, i.e., by generating a version of the program that cannot be reversed easily [18], there are also strategies to obfuscate data [23]. The main issue with obfuscation, however, is that compared to cryptography the strength of the obfuscation is typically not measurable [18]. Furthermore, these techniques target normal IT-environments and do not take the requirements of industrial systems, e.g., performance-wise, into account. In addition, they target non-collaborative environments where one side is seen as passive user, which is completely different from partners in a cooperative development environment.

Data leak detection through watermarking follows a completely different concept. While defensive mechanisms like obfuscation and anonymization strife to remove the value of the leaked information to the attacker (proactive measures), watermarking is a reactive measure that aims at proving data exfiltration in order to support legal measures. One of the main reasons is that even anonymized data might still hold significant value and it might not be possible to reduce the quality of information distributed to cooperating experts without destroying the cooperation. Typical approaches in the literature aim at protecting large amounts of structured data [24], e.g., records in a medical database. Many of these approaches work by inserting additional data, e.g. by inserting new or changing existing tuples [1] or by using the intrinsic features of the anonymization process [13]. Still, these approaches do not work for the full range of information that needs protection in industrial environments, especially not for control or configuration information, which is often only of very small size and very sensitive to even minor changes or loss of granularity as introduced by the types of algorithms outlined above.

4 Conclusion

In this work we discussed the major issues of providing security in cyber-physical production systems. The focus of our analysis was directed into two major directions, (i) the problem of introducing security into CPPS and the differences to securing traditional IT-systems and (ii) introducing security into the production

system engineering process (PSE). Especially for the latter, we not only discussed the issue of designing secure systems within an agile process, but we also discussed on how to secure the process itself with respect to manipulation and industrial espionage. Both topics have been thoroughly neglected by academics until now, but need to be considered as important issues when discussing the hardening of critical systems.

In conclusion, the topic of providing in-depth security at a professional level, comparable to the one achievable in traditional software environments, in cyber-physical production system still requires large efforts on the academic, but also the industrial side. While in this work we focused on academic research questions, it must also be kept in mind that the work to adapt the results of such research into real-life applications requires a lot of effort from the side of the industry. With dwindling profits in both, the producing industries, as well as in the production systems engineering industry, this will remain a problem. Still, as many of these cyber-physical systems also represent critical national infrastructure, solutions for securing these systems need to be devised and implemented soon.

Acknowledgements The research was funded by COMET K1, FFG - Austrian Research Promotion Agency.

References

1. Rakesh Agrawal and Jerry Kiernan. Watermarking relational databases. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 155–166. VLDB Endowment, 2002.
2. Bazara IA Barry and H Anthony Chan. Intrusion detection systems. *Handbook of information and communication security*, pages 193–205, 2010.
3. M. Barth, S. Biffl, R. Drath, A. Fay, and Winkler D. Bewertung der offenheit von engineering-tools. *open automation*, 4(13):12–15, 2013.
4. Eric Byres. The air gap: Scada's enduring security myth. *Communications of the ACM*, 56(8):29–31, 2013.
5. Michael Crosby, Pradan Pattanayak, Sanjeev Verma, and Vignesh Kalyanaraman. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2:6–10, 2016.
6. Ozgur Depren, Murat Topallar, Emin Anarim, and M Kemal Ciliz. An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert systems with Applications*, 29(4):713–722, 2005.
7. Johannes Diemer. Sichere industrie-4.0-plattformen auf basis von community-clouds. In *Handbuch Industrie 4.0 Bd. 1*, pages 177–204. Springer, 2017.
8. Interinstitutional File. Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation), 2012.
9. Eduard Fosch Villaronga, Peter Kieseberg, and Tiffany Li. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Security and Law Review*, 8 2017.

10. Kagermann Henning. Recommendations for implementing the strategic initiative industrie 4.0. 2013.
11. Michael Howard and Steve Lipner. *The security development lifecycle*, volume 8. Microsoft Press Redmond, 2006.
12. Lorenz Hundt and Arndt Lüder. Development of a method for the implementation of interoperable tool chains applying mechatronical thinkinguse case engineering of logic control. In *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, pages 1–8. IEEE, 2012.
13. Peter Kieseberg, Sebastian Schrittwieser, Martin Mulazzani, Isao Echizen, and Edgar Weippl. An algorithm for collusion-resistant anonymization and fingerprinting of sensitive microdata. *Electronic Markets*, 24(2):113–124, 2014.
14. Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.
15. Gaoqi Liang, Steven R Weller, Junhua Zhao, Fengji Luo, and Zhao Yang Dong. The 2015 ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems*, 32(4):3317–3318, 2017.
16. Udo Lindemann. *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden*. Springer, 2006.
17. Gary McGraw. Software security. *IEEE Security & Privacy*, 2(2):80–83, 2004.
18. Jasvir Nagra and Christian Collberg. *Surreptitious software: obfuscation, watermarking, and tamperproofing for software protection*. Pearson Education, 2009.
19. Ashwin Ramaswamy, Sergey Bratus, Sean W Smith, and Michael E Locasto. Katana: A hot patching framework for elf executables. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, pages 507–512. IEEE, 2010.
20. VDI Richtlinie. 2206: Entwicklungsmethodik für mechatronische systeme. *VDI-Verlag, Düsseldorf*, 2004.
21. VDI Richtlinie. 2221 (1993): Methodik zum entwickeln und konstruieren technischer systeme und produkte. *VDI-Verlag, Düsseldorf*, 2007.
22. Andreas Riel, Christian Kreiner, Georg Macher, and Richard Messnarz. Integrated design for tackling safety and security challenges of smart products and digital manufacturing. *CIRP Annals-Manufacturing Technology*, 2017.
23. Sebastian Schrittwieser, Stefan Katzenbeisser, Johannes Kinder, Georg Merzdovnik, and Edgar Weippl. Protecting software through obfuscation: Can it keep pace with progress in code analysis? *ACM Computing Surveys (CSUR)*, 49(1):4, 2016.
24. Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Watermarking relational databases. 2002.