Incorporating Attacks Modeling into Safety Process*

Amer Šurković[1], Džana Hanić[1], Elena Lisova[1], Aida Čaušević[1], Kristina Lundqvist[1], David Wenslandt[2], and Carl Falk[2]

[1] Mälardalen University, Västerås, Sweden
{asc17003, dhc17002}@student.mdh.se and
{elena.lisova, aida.causevic, kristina.lundqvist}@mdh.se
[2] Knightec AB, Sweden
{david.wenslandt, carl.falk}@knightec.se

Abstract. Systems of systems (SoS) are built as a collection of systems capable of fulfilling their own function, as well as contributing to other functionalities. They are expected to increase production efficiency and possibly decrease human involvement in harmful environments, and in many cases such systems are safety-critical. For SoS it is a paramount to provide both safety and security assurance. It is not sufficient to analyze and provide assurance of these properties independently due to their mutual connection. Hence, a joint effort addressing safety and security that provides joint guarantees on both properties, is required. In this paper we provide a safety and security assurance argument by incorporating an adversary point of view, and identify potential failures coming from the security domain that might lead to an already identified set of hazards. In this way system assets, vulnerabilities and ways to exploit them can be assessed. As an outcome mitigation strategies coming from security considerations can be captured by the safety requirements. The approach is illustrated on an autonomous quarry.

1 Introduction

Advances in operational and industrial technologies accelerate progress in the area of autonomous system of systems (SoS). SoS are built as a collection of interconnecting systems with cooperation capabilities and sharing resources allowing to extend its collective functionality, increase efficiency compared to traditional systems and provide better performance. SoS are applicable in different domains such as nuclear power plants, automotive, automation, construction works, etc. Many of such systems are safety-critical, i.e., their failure can bring harm to humans, environment or a significant money loss. Given the complexity level of SoS, their analysis with respect to safety and security arises as a paramount challenge to address.

* This work is supported by the SAFSEC-CPS project funded by KKS, the SeCRA project funded by Vinnova and the Serendipity project funded by SSF.

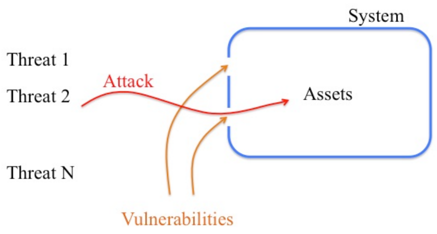

Fig. 1: Security terminology [19]

Traditionally safety and security analyses have been conducted independently, resulting in their own techniques, terminologies, standards and practices. The need for their joint consideration due to openness and interconnections of modern systems has already been recognized for more than 25 years [5] and based on the current state-of-the-art it is widely accepted in these communities. However, the state-of-the-practice on joint consideration of these properties does not have the same level of maturity yet. SoS might have external and inter-connections via modern communication infrastructures, e.g., cloud, which represent an attack surface potentially affecting system safety. Thus, to be able to guarantee such critical system properties as safety and security they need to be addressed in a joint effort.

Safety-critical systems are usually developed according to domain specific safety standards which are required to be followed for assurance purposes, as a product has to be sufficiently safe to be accepted at the market. A security breach can lead to an already identified or a new hazard, and therefore security causes leading to hazards need to be considered in order to claim a specific system safety level. A system certified to be acceptably safe without considering security related failures, can be still unsafe due to attacks potentially leading to hazards [10]. Hence, we advocate security informed safety process for autonomous SoS as for systems prone to attacks.

Consideration of safety and security in a joint effort facilitates their joint assurance. First, in Section 2 we present necessary definitions and background information related to this topic. We also recognize the necessity to identify attack models relevant for SoS, as surveyed in Section 3, and propose to connect them with a set of safety requirements in Section 4, in order to capture safety relevant security aspects as well. Thus, this paper contribution is an approach of incorporating attack models into existing safety process. As it is shown in Section 5, we complement the process with corresponding arguments presented using a goal-structuring notation (GSN) over the example of an autonomous quarry being acceptable safe, given existing threats that can jeopardize system safety. Finally, Section 6 concludes the paper.

2 Background

This section presents security terminology used in the proposed approach. **Security** can be defined as a system property allowing it *"to perform its mission* *or critical functions despite risks posed by threats"* [16], where a **threat** can be defined as *"the potential source of an adverse event"* [16].

Each system has a set of **assets**, i.e., values that need to be protected against an adversary. A **vulnerability** is a flaw in the system that enables a threat targeting one of the system assets. An **attack** realizes a threat by exploiting a vulnerability in an attempt to break a system asset as it is demonstrated in Fig. 1. **Countermeasures** are *"actions, devices, procedures, or techniques that meet or oppose (i.e., counters) a threat, a vulnerability, or an attack by eliminating or preventing it"* [16]. They can be classified as (*i*) preventive, e.g., encryption, (*ii*) detective, e.g., intrusion detection systems, (*iii*) responsive, e.g., forensics [22]. An **attack model** can be defined as an instantiation of an adversary model in a specific scenario [28], where the latter implies adversary capabilities, constraints and possible interactions with the system. Thus, an attack model demonstrates how an adversary can achieve his or her goal by different techniques and methods for launching an attack, which threats are realized, which vulnerabilities are exploited and which assets are targeted [27].

3 An Overview of Existing Attack Models

In this section we present a summary of the literature survey on attack models [9]. The survey papers from 2010 - 2018 in the following databases: IEEE Explore Digital Library, Springer Link, Web of Science and ACM. The identified papers have been grouped according to the application domain. The majority of selected papers, (10) are originated from control systems domain followed by vehicular and recommended systems domain, (6) and (5) papers correspondingly, whereas IoT and cloud computing is the least represented. The latter can be justified by the relative novelty of the domains.

Control systems can be categorized into Process Control Systems, Supervisory Control and Data Acquisition (SCADA) Systems, Distributed Control Systems and Cyber Physical Systems (CPSs). They are becoming increasingly vulnerable as they are more exposed and available towards open networks. Existing attack models are focused either on general problems like protocols in SCADA systems, or on specific problems, i.e., smart grid subsystems [27]. The identified attack models are a general sensor attack model from which Denial of Service (DoS) and integrity attacks can be launched [4], attack models in CPSs that can be summarized into DoS and deception attacks [17], aspect-oriented models for CPSs [33], an attack model for CPSs instantiated for a Secure Water Treatment (SWaT) system [1], a smart grid attack model [21], attack models tackling the sparsity of attacks in a distributed smart grid framework [26], and data injection attack models that target integrity of sensor measurements for power grid systems [23].

Attack models related to vehicular domain are exploiting in-vehicle control area network (CAN) vulnerabilities, vulnerabilities in on-board units (OBU) [31], an electric vehicle infrastructure [24]. They might also provide a specific attack, e.g., an attack on a vehicle position forging attacks [8], or specific vehicle type, e.g., resource-constrained Unmanned Aerial Vehicle (UAV) [14]. In IoT attack models can be related to its middleware [7] or be focused on a particular attacks, e.g., the command disaggregation attack [35]. In the context of cloud services, attack models might be connected with an issue of service providers getting access to sensitive client information [36], or aligned with stages of using cloud services, i.e., registration, data gathering about the infrastructure and finally creation of virtual machines for accessing data from other clients [3].

Attack models targeting communication part of systems might be categorized as attack models based on targeted functionalities in different network layers [29], or grouped based on an attack goal as DoS and deception attacks [6]. Further, they could aim for specific protocols, e.g., HTTP/2 Internet service [2], or a specific attack, e.g., jamming attacks for wireless networks [34]. Considering radio-frequency identification (RFID) applications the following attack models have been identified: a forgery attack, a replay attack, a man-in-the-middle attack, a tracking attack [20], DoS, an eavesdropping and scanning [15] and, finally, those attacks focusing on air interfaces [25]. The last identified area for attack models is recommender systems, systems that try to predict a user preference based on the previous behavior of the user, where a large number of new web-services makes it difficult to maintain quality of service for clients [18]. The majority of publications in this area are focused on shilling attacks [13, 32, 37], i.e., an attack with the goal to manipulate the output of a recommender system. However, other types of attacks, e.g., injection attacks [11], are also used for attack models in recommended systems.

4 Attack Models and Safety Process

4.1 Inclusion of Attack Models into Safety Process

This work that combines attack models with functional safety requirements is an extension of our initial idea of incorporating security concerns into safety process [30]. Fig. 2 depicts a reference structure of a safety process, where based on a given system definition, hazard analysis and risk assessment are conducted followed by formulation of safety goals and elicitation of corresponding functional and technical safety requirements (FSRs and TSRs). By executing the system development process artifacts are collected and used as an input into a security analysis. We propose to engage the attack modeling process once there is enough artifacts collected, i.e., hazards and safety goals are formulated. The approach allows engaging the attack modeling process on demand, e.g., if there is an update in the system and correspondingly during the artifacts collection, addressing the dynamic nature of security. The attack modeling process starts with identification of system assets and iterates later on for each identified asset. Each iteration includes identification of related system vulnerabilities, risk assessment of potential threats and finally identification of possible attacks targeting the considered asset. The output of the process is a set of mitigation techniques or countermeasures that is forwarded as an input to the FSRs elicitation step.

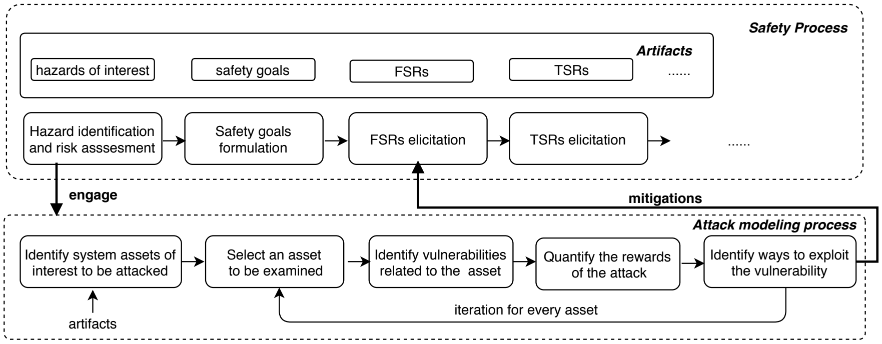
Safety Process
Artifacts
hazards of interest
safety goals
FSRs
TSRs
......
Hazard identification and risk asssesment
Safety goals formulation
FSRs elicitation
TSRs elicitation
......
engage
mitigations
Attack modeling process
Identify system assets of interest to be attacked
Select an asset to be examined
Identify vulnerabilities related to the asset
Quantify the rewards of the attack
Identify ways to exploit the vulnerability
artifacts
iteration for every asset

Fig. 2: An approach of incorporating attack modeling into safety process

4.2 Use Case: An Autonomous Quarry

We illustrate our approach on an example of an autonomous quarry. The quarry is equipped with a battery-powered electric load carriers capable to cooperate with other machines such as wheel loader. They are expected to follow a path, load/unload, transport, avoid waiting and carrying load over longer distances than needed, as well as any unnecessary movement, including rework. The goal is that a fleet of these unmanned carriers is jointly able to move the same amount of load as one large haul truck and in case any of these carriers would go down, the loss to the overall quarry production should be much smaller, compared to the loss of a large haul truck. Assuming the carriers being fully autonomous, all possible processes and scenarios need to be documented and analyzed, taking into consideration all new critical situations, including possible threats coming from the security domain affecting the safety of the system. The described autonomous quarry follows the ISO 17757 standard [12] to document safety requirements and criteria for semi-autonomous and autonomous machines and associated systems, typically used in earth-moving and mining operations. For a given use case we have been provided with a quarry architecture description and a list of hazards, identified based on ISO 17757.

Given our findings in Section 3, we have chosen to work with an attack model described by Wang et al. [31] that focuses on the in-vehicle network and ways to compromise it. In general, in-vehicle networks are considered as closed networks and secure from malicious attacks, but with multiple network access (e.g., PC, co-pilot unit), there is a number of threats that might endanger them (e.g., current OBUs used in vehicles fail to protect network due to the lack of awareness of possible attacks). Also, an attacker may perform illegitimate vehicle control through unsecured OBUs and in-vehicle CAN. For the in-vehicle CAN the following vulnerabilities have been identified: *(i)* weak access control mechanism, *(ii)* CAN data frames do not have encryption, and *(iii)* no authentication in data exchange exists.

We have chosen two scenarios, that are **short-range attack** and **long-range attack**. Wang et al. [31] describe two methods for short-range attack. In the first, attackers camouflage as a legitimate user device through the same communication protocol derived from stolen data that allows them to send illegitimate control commands to the in-vehicle CAN. In the latter, attackers may develop and implement security protocols on their own that is possible due to the direct communication between external devices and in-vehicle CAN. Furthermore, the following attacks can be derived for the selected attack model: *(A1)* **a forgery attack** that communicates with braking system using commands as a legitimate user device or an OBU; *(A2)* **a DoS attack** resulting in information blocking by injecting irrelevant data into in-vehicle CAN and OBU; *(A3)* **a replay attack** affecting operation of braking equipment by repeatedly transmitting data to CAN; *(A4)* **an eavesdropping attack** resulting in stealing users data and compromising privacy. Described attacks might be counter-reacted using the following security measures: *(M1)* the identity authentication or access control; *(M2)* data authentication and filtering false information; *(M3)* blocking a large number of packets; *(M4)* hardware isolation.

4.3 Hazards of Interest for the Presented Attack Model

Based on the provided documentation, we have selected a set of hazards of interest for this work [9]. However, in this paper we present information about only one hazard detailed below to illustrate the approach.

The navigation and collision hazard due to: *(i)* failures to detect in time an object; *(ii)* increased latency caused by other applications or computation loading to the processor being used for the object detection or classification system; *(iii)* material on the transmitter or receiver erroneously detected as objects; *(iv)* erroneous location of a detected object; *(v)* inability to stop the machine remotely or in an emergency state; *(vi)* lack of access to situational awareness information; *(vii)* inaccurate terrain data; *(viii)* lost or delayed command input; *(ix)* inaccurate position (due to loss of GNSS correction); *(x)* inaccurate planning information; *(xi)* incomplete or improper system updates and changes to software; **caused by either a DoS attack or a forgery attack**.

5 Joint Safety and Security Argumentation

5.1 GSN for the Autonomous Quarry

We aim to identify possible attacks provided that an attack model exist and discover which of them can cause already recognized hazards. Due to the space limitation, we choose to present only one hazard. Fig. 3 depicts a part of the argument for the chosen hazard *"joint navigation and collision hazard (H_1)"*. In the presented argument, we take into account a possibility that a DoS attack on in-vehicle CAN might occur in the quarry since it has enabled Internet connection, blocking the information transmission on-board the vehicle from the main vehicle processor to sensors and/or ECUs that it communicates to. The communication that is performed through CAN is safety-critical since it occurs in real time and failure to obtain an information from an expected ECU might

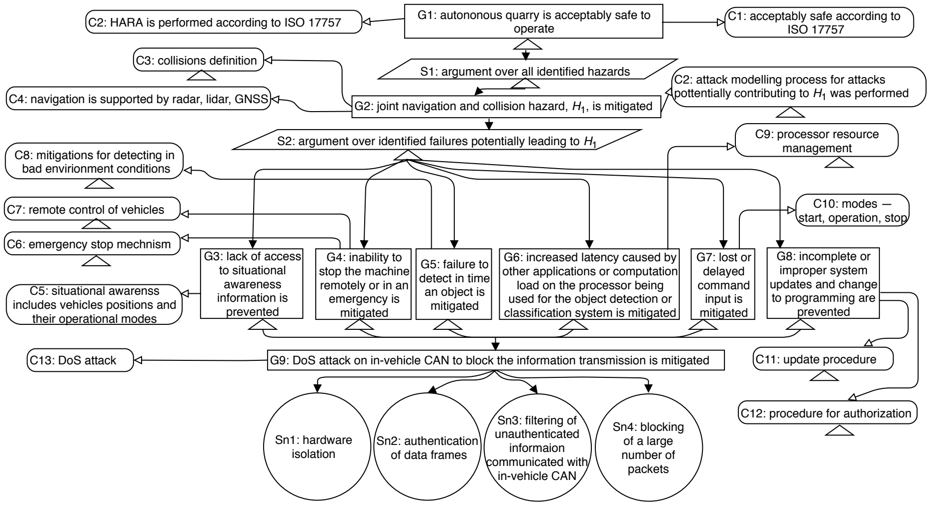
G1: autononous quarry is acceptably safe to operate
C2: HARA is performed according to ISO 17757
C1: acceptably safe according to ISO 17757
S1: argument over all identified hazards
C3: collisions definition
C4: navigation is supported by radar, lidar, GNSS
G2: joint navigation and collision hazard, H_1, is mitigated
C2: attack modelling process for attacks pottentially contributing to H_1 was performed
S2: argument over identified failures potentially leading to H_1
C9: processor resource management
C8: mitigations for detecting in bad envirionment conditions
C7: remote control of vehicles
C10: modes — start, operation, stop
C6: emergency stop mechnism
C5: situational awarenss includes vehicles positions and their operational modes
G3: lack of access to situational awareness information is prevented
G4: inability to stop the machine remotely or in an emergency is mitigated
G5: failure to detect in time an object is mitigated
G6: increased latency caused by other applications or computation load on the processor being used for the object detection or classification system is mitigated
G7: lost or delayed command input is mitigated
G8: incomplete or improper system updates and change to programming are prevented
C13: DoS attack
G9: DoS attack on in-vehicle CAN to block the information transmission is mitigated
C11: update procedure
C12: procedure for authorization
Sn1: hardware isolation
Sn2: authentication of data frames
Sn3: filtering of unauthenticated informaion communicated with in-vehicle CAN
Sn4: blocking of a large number of packets

Fig. 3: Argument for the hazard H_1

lead to a hazard. Having that in mind, performing a DoS attack on quarry's autonomous vehicles might contribute to failures leading to (H_1) as described in Section 4.3. Therefore, we have introduced seven sub-goals, $G2$- to prevent this.

A DoS attack might cause *lack of access to situational awareness information* captured by $G3$, as a vehicle would not be able to gain real time information from its sensors regarding the surrounding environment and its position. This might lead to uncontrolled vehicle movements at the quarry creating and possibly endangering high value equipment at the site, including itself, and even cause a risk to people at the quarry. In case of the *inability to stop machine remotely or in an emergency situation* failure, captured by $G4$, not being prevented and the in-vehicle CAN being flooded with DoS information packets, a vehicle would not be able to perform safety-critical functions such as emergency stop. This is one of the highest degree severity attacks since it may block one of the core safety functions of the vehicle.

The failure *to detect or late detection of an object* addressed by $G5$, that can be caused by communication between modules sending important commands and information within a vehicle, is either limited or completely disabled due the DoS attack. A vehicle would not be able to react to critical situations such as avoiding obstacles. *Increased latency in system functions due to other applications or computation load* failure addressed by $G6$ if not mitigated might reduce the overall system performance. If such event is introduced to the processor used for the object detection, severity of the attack would increase. With the DoS attack, the failure *command inputs can be either lost or delayed* captured by $G7$, can be introduced to the system. Communication channels might be blocked with sufficient amount of irrelevant data packets, causing command inputs to either be lost in the transmission or delayed long enough for a hazard to occur. This might completely stop the operation at the quarry. Moreover, the DoS attack on in-vehicle CAN may cause *incomplete or improper system updates* failure, addressed by $G8$, causing major disturbance in the functioning of the quarry or delays in performance and introducing potential financial losses.

The proposed mitigation/prevention strategies for the described hazard are selected based on existing findings in regard to DoS attacks [31], that are *(i)* hardware isolation (Sn1), *(ii)* authentication of data frames (Sn2), *(i)* filtering of unauthenticated information communicated with in-vehicle CAN (Sn3), and *(iii)* blocking of a large number of packets (Sn4).

5.2 Incorporating Security in Argumentation over Safety

One of the steps towards joint assurance of safety and security is development of an argument structure to support it. In the example of an autonomous quarry, security consideration brought in additional solutions that need to be captured in the corresponding requirements. However, this might require changes in patterns of arguments itself, as it is not enough to argue over system vulnerabilities being mitigated, as security is dynamic and one may also require to argue over system patches being implemented timely in place due to established security process. In this work a security assurance has been introduced at the stage when parts of the safety assurance have been already done (i.e., safety requirements elicited, hazard analysis and risk assessment conducted, etc.). However, a joint assurance assumes security being considered during system development process as different phases of development may require different levels of assurance.

The most important difference between arguing security compared to arguing safety of a system is the presence of an adversary. The behavior of adversaries is not predictable, implying that security threats evolve and adapt with time and therefore an existing case might have its assumptions unexpectedly being violated, or its strength might not be adequate to protect against new attacks. Therefore assurance cases would need to be revisited more frequently than assurance cases covering only safety. Based on this, system assets to be protected change and new vulnerabilities arise. The system evolution of that kind goes against the static structure of an assurance case that in this example would requires (re-)building the case from scratch given any update at run-time. Therefore, it is crucial to enable continuous assurance through the entire life cycle and provide arguments regarding evolving assets and mitigation actions on new vulnerabilities. This process can be seen as a way of enabling run-time assurance of systems that evolve over time (e.g., self-adaptive systems). Run-time assurance case adaptation would not only allow handling of updates in a cost-efficient and effective way, but would be able to facilitate continuous joint assurance of systems that adapt at run-time.

6 Conclusions

Well established methods, techniques and processes within separate communities of safety and security are not sufficient anymore to produce acceptably safe and secure systems, most importantly they should not be isolated one from another.

With the growing number of cyber attacks on safety-critical systems, we have identified the need to observe a system from an adversary point of view. In this paper we choose to incorporate an attack that focuses on ways to compromise an in-vehicle network and include the knowledge about preventing/mitigating it while providing argumentation for system safety. We demonstrate parts of the argumentation at example of autonomous quarry using GSN. In the future we plan to investigate ways how this information can be included into security assurance case, similar to one from safety domain, possibly at run-time.

References

1. Adepu, S., Mathur, A.: An Investigation into the Response of a Water Treatment System to Cyber Attacks. In: 17th IEEE International Symposium on High Assurance Systems Engineering (2016)
2. Adi, E., Baig, Z.A., Hingston, P., Lam, C.P.: Distributed denial-of-service attacks against http/2 services. Cluster Computing (2016)
3. AlJahdali, H., Albatli, A., Garraghan, P., Townend, P., Lau, L., Xu, J.: Multi-tenancy in cloud computing. In: 8th IEEE Int. Symposium on SOSE (2014)
4. Cárdenas, A.A., Amin, S., Lin, Z.S., Huang, Y.L., Huang, C.Y., Sastry, S.: Attacks Against Process Control Systems: Risk Assessment, Detection, and Response. In: ACM Symposium on Information, Computer and Communications Security (2011)
5. Causevic, A.: A risk and threat assessment approaches overview in autonomous systems of systems. In: The 26th IEEE International Conference on Information, Communication and Automation Technologies (2017)
6. Ding, D., Wang, Z., Wei, G., Alsaadi, F.E.: Event-based security control for discrete-time stochastic systems. IET Control Theory Applications (2016)
7. Ferreira, H.G.C., de SousaJunior, R.T.: Security analysis of a proposed internet of things middleware. Cluster Computing (2017)
8. Grover, J., Laxmi, V., Gaur, M.S.: Attack models and infrastructure supported detection mechanisms for position forging attacks in vehicular ad hoc networks. CSI Transactions on ICT (2013)
9. Hanić, D., Šurković, A.: An Attack Model of Autonomous Systems of Systems. Master's thesis, Mälardalen University, IDT (June 2018)
10. Hänninen, K., Hansson, H., Thane, H., Saadatmand, M.: Inadequate risk analysis might jeopardize the functional safety of modern systems (March 2016)
11. Huang, S., Shang, M., Cai, S.: A hybrid decision approach to detect profile injection attacks in collaborative recommender systems. In: Foundations of Intelligent Systems. Springer Berlin Heidelberg (2012)
12. ISO 17757 - International Organization for Standardization: Earth-moving machinery and mining-, and semi-autonomous machine system safety (2017)
13. Jiang, F., Tian, R.: The influence of shilling attacks with different attack cycles. In: 6th IIAI International Congress on Advanced Applied Informatics (2017)
14. Katewa, V., Anguluri, R., Ganlath, A., Pasqualetti, F.: Secure reference-tracking with resource-constrained uavs. In: IEEE CCTA (2017)
15. Khan, G.N., Yu, J., Yuan, F.: Xtea based secure authentication protocol for rfid systems. In: ICCN (2011)
16. Kissel, R.: Glossary of key information security terms. U.S. Dept. of Commerce, National Institute of Standards and Technology (2006)
17. Kwon, C., Liu, W., Hwang, I.: Security analysis for cyber-physical systems against stealthy deception attacks. In: American Control Conference (June 2013)

18. Li, X., Gao, M., Rong, W., Xiong, Q., Wen, J.: Shilling attacks analysis in collaborative filtering based web service recommendation systems. In: IEEE International Conference on Web Services (2016)
19. Lisova, E.: Monitoring for Securing Clock Synchronization. Ph.D. thesis, Mälardalen University (April 2018)
20. Liu, H., Ning, H.: Zero-knowledge authentication protocol based on alternative mode in rfid systems. IEEE Sensors Journal (2011)
21. Lu, Z., Wang, W., Wang, C.: Camouflage Traffic: Minimizing Message Delay for Smart Grid Applications under Jamming. IEEE Transactions on Dependable and Secure Computing (2015)
22. Miede, A., Nedyalkov, N., Gottron, C., Knig, A., Repp, N., Steinmetz, R.: A Generic Metamodel for IT Security Attack Modeling for Distributed Systems. In: International Conference on Availability, Reliability and Security (2010)
23. Mohammadi, A., Plataniotis, K.N.: Secure estimation against complex-valued attacks. In: IEEE Statistical Signal Processing Workshop (2016)
24. Mousavian, S., Erol-Kantarci, M., Wu, L., Ortmeyer, T.: A risk-based optimization model for electric vehicle infrastructure response to cyber attacks. IEEE Transactions on Smart Grid (2017)
25. Ning Huansheng, L.H., Chen, Y.: Ultralightweight rfid authentication protocol based on random partitions of pseudorandom identifier and pre-shared secret value. Chinese Journal of Electronics (2011)
26. Ozay, M., Esnaola, I., Vural, F.T.Y., Kulkarni, S.R., Poor, H.V.: Distributed models for sparse attack construction and state vector estimation in the smart grid. In: 3rd IEEE International Conference on Smart Grid Communications (2012)
27. Paudel, S., Smith, P., Zseby, T.: Attack Models for Advanced Persistent Threats in Smart Grid Wide Area Monitoring. In: 2nd CPSR-SG. ACM (2017)
28. Rocchetto, M., Tippenhauer, N.O.: On attacker models and profiles for cyber-physical systems. In: Computer Security – ESORICS 2016. Springer (2016)
29. Sunghyuck, H., Sunho, L., Jaeki, S.: Unified modeling language based analysis of security attacks in wireless sensor networks: A survey. KSII Transactions on Internet and Information Systems (2011)
30. Surkovic, A., Hanic, D., Lisova, E., Causevic, A., Wenslandt, D., Falk, C.: Towards attack models in autonomous sos. In: IEEE SoS Engineering (2018)
31. Wang, L., Liu, X.: NOTSA: Novel OBU with Three-level Security Architecture for Internet of Vehicles. IEEE Internet of Things Journal (2018)
32. Wang, Y., Wu, Z., Cao, J., Fang, C.: Towards a tricksy group shilling attack model against recommender systems. In: Zhou, S., Zhang, S., Karypis, G. (eds.) Advanced Data Mining and Applications. Springer Berlin Heidelberg (2012)
33. Wasicek, A., Derler, P., Lee, E.A.: Aspect-oriented modeling of attacks in automotive cyber-physical systems. In: 51st ACM/EDAC/IEEE DAC (2014)
34. Xu, W., Trappe, W., Zhang, Y., Wood, T.: The feasibility of launching and detecting jamming attacks in wireless networks. In: 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (2005)
35. Xun, P., Zhu, P.D., Hu, Y.F., Cui, P.S., Zhang, Y.: Command Disaggregation Attack and Mitigation in Industrial Internet of Things. Sensors (2017)
36. Yiu, M.L., Ghinita, G., Jensen, C.S., Kalnis, P.: Enabling search services on outsourced private spatial data. The VLDB Journal (2010)
37. Zhang, F.: Analysis of bandwagon and average hybrid attack model against trust-based recommender systems. In: 5th ICMeCG (2011)