Clock Synchronization Considerations in Security Informed Safety Assurance of Autonomous Systems of Systems

Elena Lisova, Aida Čaušević, Elisabeth Uhlemann, Mats Björkman
Mälardalen University, Västerås, Sweden
elena.lisova@mdh.se, aida.causevic@mdh.se, elisabeth.uhlemann@mdh.se, mats.bjorkman@mdh.se

***Abstract*—Over the past decade, fast technological and industrial advances have been happening in the area of autonomous Systems of Systems (SoS). A SoS is built upon integration of several systems, where the complexity of such a structure is exponentially higher which brings challenges to its analysis. However, it also has provided a large set of new opportunities in domains such as air traffic control, defense, construction industry, etc. It is expected that fully autonomous and cooperating systems will increase the production efficiency, while decreasing (potentially completely replacing) the human effort in harmful environments. In order to enable this, we need to make sure that critical properties of SoS, such as safety and security are guaranteed. We believe that it is not sufficient anymore to analyze and guarantee these properties independently, but we have to be able to address safety and security in a joint effort.**

Communications in systems with any type of real-time requirements, where data validity is based on its freshness, rely on clock synchronization (CSyn) allowing its subsystems to cooperate and work coherently. Considering reliable and predictable communication as one of the main assets contributing to correct functionality of such systems, protecting CSyn from malicious adversaries should be one of the highest priority efforts in SoS. In this paper we show how CSyn breaches can influence security, and ultimately safety of complex and autonomous SoS, further we identify a missing piece to consider in safety assurance, namely assurance with respect to reliable communications between systems within the SoS. We demonstrate how an outcome of a security analysis can be used as input for the overall safety analysis and we use an autonomous quarry as an example application to illustrate our findings.

I. Introduction

Systems around us are no longer observed as separate units, but as a part of larger cooperating systems, connected to public or semi-public networks, where information errors can propagate throughout the system in many, sometimes unpredictable ways. The system control no longer depends on human operators only, but also on other systems they are connected to. They involve multiple stakeholders, have dynamic system reconfigurations, and unpredictable operating environments. Such software-intensive systems are also referred to as Systems of Systems (SoS).

To enable dependability guarantees in systems like this, both safety and security have to be satisfied through dedicated efforts in detecting and recovering from failures. Historically, safety and security have been addressed by two distinct communities, each focusing on their own methodologies, techniques and tools for system development. However, already in the 1990s, some researchers noticed commonalities between these dependability properties [1]–[5] and tried to provide a way to reason about them in a unified way. There have been several efforts to (re)use already existing techniques, trying to identify similarities, as well as differences, introducing perspectives on how the properties of safety and security can be harmonized [6]–[11]. Given the existing literature, one can notice that the interdependencies between safety and security are increasingly understood and accepted, however there is still a lack of dedicated methods and approaches that consider safety and security jointly while designing and developing software-intensive systems.

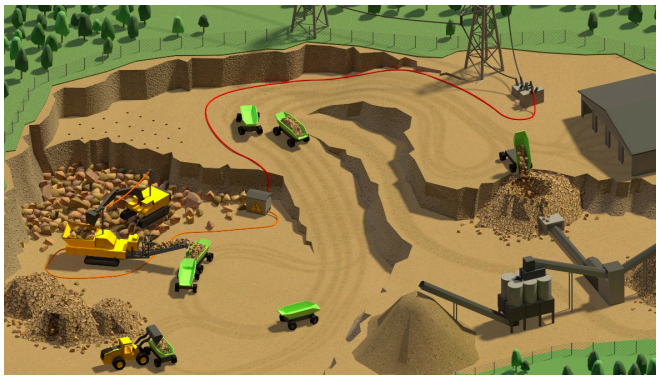

Fig. 1. An example of an autonomous quarry [12]

Let us consider a fully autonomous quarry as depicted in Fig. 1, where battery-powered electric load carriers operate in cooperation with other machines. These carriers are expected to follow a path, load/unload, transport, avoid waiting, carrying load over longer distances than needed and any unnecessary movements including rework. It is expected that a fleet of these unmanned carriers will jointly be able to move the same amount of load as one large haul truck and if one 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. However, since these machines are assumed to be fully autonomous, all possible processes and scenarios need to be documented and analyzed, taking into consideration all new critical situations. Considering different types of communication in such systems (e.g., GPS, machine to server communication, machine to machine communication,

etc.), one has to account for all possible threats coming from the security domain affecting the safety of the system, as well. For instance, just by delaying message delivery, one could provoke unexpected events, such as carrier crashes, carriers being unavailable at the expected time, most likely causing loss of production and decreasing the overall efficiency.

In complex SoS as autonomous quarries are, one has a set of assets to protect which also includes communication and protection of clock synchronization (CSyn) as these are paramount requirements for proper scheduling of message exchange [13]. In these systems it is important to be able to keep track of fresh data [14] and assess data validity based on the time it has been obtained. Furthermore, if the CSyn is breached the whole network becomes disturbed. Therefore, securing CSyn is an emerging and important challenge to consider when enabling reliable and predictable communication in complex SoS. Moreover, it is evident that the security level of CSyn affects the SoS safety, as broken CSyn can lead to potential hazards related to system correctness, availability and reliability.

In this paper we explore interdependencies between safety and security given that threats related to CSyn exist. We show how CSyn protection can influence the safety of complex and autonomous SoS and identify a missing piece to consider in safety assurance, namely assurance with respect to predictable communications between systems within the SoS. The scope of the approach is SoS with a time-triggered architecture. The findings are illustrated with Goal Structuring Notation (GSN), a notation commonly used in safety assurance, in a form of a possible argument regarding CSyn. Furthermore, we use the example of an autonomous quarry to demonstrate the identified challenges and interdependencies.

The paper is organized as follows. In Section II we present the necessary background information regarding safety, security, and CSyn. We describe our example in Section III, whereas CSyn issues in software-intensive systems are presented in Section IV. In Section V, we describe our approach for joint safety and security assurance with respect to the communication in SoS. Finally, we conclude the paper with Section VI.

II. Background

In this section we briefly present security and safety concepts used in this paper, an assurance case representation through GSN, and the notion of CSyn.

A. Security and Threat Analysis

Security can be defined as a system property that allows it "to perform its mission or critical functions despite risks posed by threat" [15]. A threat in its turn can be defined as "the potential source of an adverse event" [15]. A vulnerability is a flow in the system that allows an adversary to realize a threat targeting one of the system assets. A concrete threat realization is an attack. Therefore, once a high-level threat has been identified based on an adversary model, knowledge of the adversary goal and existing vulnerabilities in the system, the threat can be decomposed into possible attacks leading to this threat.

Attack Tree Analysis (ATA) is a formal technique used in security for threat modeling [16]. The root of such a tree is an adversary goal corresponding to an identified high-level threat. The root has leafs that presents possible ways of achieving the root, i.e., possible threats. Leafs can be connected via AND or OR gates. Each leaf can be decomposed further until the desired level of granularity is achieved.

B. Safety and Fault Analysis

Safety can be defined as "freedom from unacceptable risk of physical injury or of damage to the health of people" [17]. A system cannot be absolutely safe, but it can be acceptably safe. To define risks, top-level events leading to an accident, i.e., hazards, should be identified. To demonstrate that risks are addressed and minimized to an acceptable level, a safety assurance case is required. A hazards identification procedure and an analysis is a part of the case.

Fault Tree Analysis (FTA) [18] is a technique to analyze events causing hazards. The root of the three is a high-level hazard that is iteratively decomposed further into single failures via AND or OR gates. Further developed intermediate, undeveloped intermediate and basic events are depicted using rectangular, rhombic and circular shapes respectively [19].

C. Security and Safety by Process Analysis

System-Theoretic Process Analysis for Security (STPA-Sec) is an approach that deals with securing software intensive systems against intentional disruptions [10], where instead of focusing on threats coming from adversary actions, one focuses on controlling system vulnerabilities. STPA-Sec follows four basic process steps: *i)* establish the systems engineering foundation for the security analysis; *ii)* identify the control actions that constitute a threat to system security; *iii)* use control actions to create security requirements and constraints; *iv)* identify scenarios in which the security constraints are violated. In general, this approach does not give any concrete information about what specific counter measures that should be taken, but provides a useful way to identify scenarios that should be the focus of security experts when securing complex systems and especially SoS.

D. Assurance Case Representation

An assurance case is required to assure that a desired system property is provided. It consists of structured arguments and evidences supporting them [20]. To avoid being ambiguous, a special unified notation is used for reasoning, namely **GSN** [21]. It consists of the following components: *a goal*, i.e., what should be achieved, *a strategy*, i.e., how it should be achieved, *a solution*, i.e., means for the goal realization, and *a context*, i.e., assumptions and conditions.

E. Clock Synchronization

In a network where nodes follow any kind of schedule, they need to have the same notion of time, e.g., to be

able to send out a message in a prescheduled time-slot. In event-triggered complex networks, nodes have to be able to judge the information freshness and validity by looking at its timestamps, and thus they also here need to share the same notion of time, i.e., being synchronized. Two nodes are synchronized when the difference between their clock times stays within specified boundaries. Each node has its own clock and a natural drift. The drift results in a clock offset. An offset is the difference in time between a node time and a reference time for it, e.g., in a grandmaster/slave approach it is the grandmaster time. This drift cannot be eliminated, thus it should be periodically corrected via CSyn algorithms, so that the clock offset stays within the given boundaries. There are different approaches for the clock correction, it can be performed by sharp correction each resynchronization interval [22], i.e., jumping to an estimated correct time, or by more smooth and monotonic corrections of the drift slope [23].

III. Autonomous Quarry

In this paper, we consider a fully autonomous quarry to illustrate and motivate our findings. It is an example of a SoS as it includes several subsystems: wheel loaders, autonomous electric carriers, rock crushers, charging stations, trucks, a remote control room, etc. (Figure 1), where each subsystem is a stand-alone system with its own function and purpose, but capable of sharing its resources and capabilities to create a new, more complex SoS. A quarry operation is organized in six main phases: *(i)* site establishment, *(ii)* exploitation, *(iii)* processing, *(iv)* distribution, *(v)* maintenance and *(vi)* reclamation [24]. Quarries are often situated in remote areas without cellular communication coverage. Additionally, the environment is harsh due to dust and solid materials that implies obstructed line-of-sight for the wireless communication. The environment and topology change over time imposing challenges on providing reliable maps, location, path and route data for the involved subsystems. Reliable communication is an important asset to enable accurate and correct decisions to be made. Otherwise any communication disruptions might lead to inaccurate actions, negatively affecting production, cost and most likely safety and security at the site. Therefore a reliable and predictable wireless link is of absolute importance. Different types of wireless communication technologies between autonomous machines are possible: *satellite* (provides good coverage, but has limited bandwidth, high latency and cost), *cellular* (comes with high bandwidth, but the coverage might be hard to ensure in remote areas) and *dedicated short range communication* such as vehicular ad hoc networks (VANETs). VANET was initially intended for use in the car industry, but there has been some efforts using it in the domain of construction equipment [25]. We assume that each construction machine broadcasts its activity (machine type, machine task, operational status) and basic position data (position, speed, and direction). Autonomous electric load carriers run on a battery and it is important to enable charging to prevent any disruptions in normal production. They are also equipped with a vision system, which allows the machine to detect humans and obstacles in its vicinity. The whole quarry can be seen as a complex cooperative SoS, where the machines should work in coherence, synchronized with each other to enable efficient operation and avoid losses in production, equipment or in the worst case a human life.

A. Identified challenges

An autonomous quarry is an open safety-critical SoS in which expected production and operation of all subsystems rely on a correct and timely exchange of messages over the communication network. Complexity of SoS brings many challenges in its analysis and evaluation in terms of safety and security [26], [27].

Vulnerabilities related to communication channels open up possibilities to breach SoS and cause new or contribute to existing safety hazards. In this paper we focus only on the security challenges emerging from the communication, as it goes beyond traditional safety analyses. We are interested in learning about malicious attacks on the CSyn mechanisms in the system, and its effect on system safety. It is a paramount to enable exchange of fresh and valid messages within given time-slots, otherwise it may lead to several hazards including loss of a production due to electric load carriers not being charged, collisions within the site given that the old message regarding the position of machines has been sent, etc.

IV. Clock Synchronization in Complex Software-Intensive Systems

Complex software-intensive SoS include different communicational protocols, e.g., TT-Ethernet, WirelessHART, as well as other protocols supporting the communications, e.g., protocols for scheduling and network reconfiguration. CSyn is usually assumed to be in place by many of these protocols, e.g., in scheduling. Therefore, an additional protocol is usually deployed to establish and maintain CSyn. There are several protocols widely used in industry, e.g., Network Time Protocol (NTP) [28], IEEE 802.1AS [29] or Precision Time Protocol (PTP) [22]. However, since the same protocol is used for many applications, an attack exploiting its vulnerability and causing CSyn breaching, can be reused by an adversary, and thus is appealing.

Traditionally, CSyn protocols do not have security solutions in place. For example, IEEE 1588 has only optional security extensions covering group authentication and countermeasures for an replay attack [30]. CSyn can be broken if the offset is calculated incorrectly or if there is no available information to calculate it. Therefore, if there is no encryption of timestamps or integrity check for synchronization messages, timestamps values can be changed and a node can bring itself into an unsynchronized state. However, even if messages are encrypted, a delay attack will still be successful [31]. In the delay attack, the message is delayed, but not modified. As an outcome, the receiver gets the time of message arrival changed without knowing it. This type of attack is one of the most challenging to detect and provide counter measures for [32]. Finally, even if the cause of a CSyn breach comes from the

security domain, it has direct implications on system safety. If a node in the unsynchronized state, cannot propagate timely alarm messages, it causes a failure. Thus, CSyn is an excellent example of safety and security overlapping each other.

In the SoS presented in Section III, there are several types of wireless communications. According to our assumptions, nodes have time-slot allocated access to the communication channel. Also, a control unit needs to assess data freshness by its timestamps to make a decision. Therefore, if an electric load carrier is brought into unsynchronized state, depending on the offset it has compared to other network participants, it can time stamp data with an incorrect time or even try to access the channel in someone else's time-slot, with resulting collisions. In the first case, a control unit outside the vehicle can make an incorrect decision based on the received data. In the second case a decision required to be made due to the change of the environment of the carries can not be provided as no information is received due to data collisions.

V. An Approach for Joint Safety and Security Assurance in Communication

To enable a joint safety and security assurance of software-intensive SoS such as the one described in Section III, we use the well known STPA-Sec approach and combine it with GSN to get a structured way of deriving safety cases. In this paper, we focus only on network communication vulnerabilities resulting in CSyn breaches. More particularly, in this work we build upon the first step of existing STPA-Sec where we establish foundations for a security analysis by formulating a threat model. This model is complemented with ATA and FTA to illustrate how threats derived from a threat model can lead to hazards for the considered use case.

A. Security and Safety Assurance w.r.t. Clock Synchronization

In order to identify high-level threats for the autonomous quarry use case, we build a threat model consisting of five blocks as depicted to the left in Fig.2. We complement the already existing structure for threat modeling of industrial systems [33] with a relevant set of vulnerabilities. The first block describes a use case and its specification. All components of the model are tightly dependent on the use case specifics and built upon it. The second block represents assets of the use case, i.e., what should be secured. There are several relevant assets required to assure data transmission, but here we focus on CSyn as it needs to be guaranteed for all use cases with any kind of real-time requirements. A remote control, enabled by the transmitted data can also be guaranteed only if a control unit can judge about information freshness based on its timestamps.

The third block represents an adversary model, i.e., a collection of assumptions about a possible adversary. We assume that the adversary might be familiar with the SoS architecture and know which protocols are used for CSyn.

Generally, adversary goals can be related to hijacking, disruption or eavesdropping as shown in the forth block. We identify disruption as the main goal when intentionally breaching a CSyn. The fifth block describes possible vulnerabilities. Since SoS, such as the autonomous quarry is, rely on communication infrastructure (i.e., satellite, cellular and/or VANET), we isolate communication as the main vulnerability that needs to be addressed in order to protect the assets of the system.

Having the threat model completed, we can derive a high-level threat related to the considered SoS asset: *a CSyn is broken for a time sufficiently long to allow the adversary to reach its goal and its breach has not been detected within the specified time, i.e., before consequences are evinced*. To analyze this threat, further ATA should be performed to identify possible single attacks realizing this threat.

The complexity of the SoS structure requires a revised taxonomy of hazards in order to identify and analyze them properly [34]. If we consider a high-level FTA, an SoS hazard can be decomposed into *single* and *emergent* hazards. Single hazards are usually related to a particular system within the SoS. Conversely, emergent hazards are the results of the integration of several systems into SoS, which is exactly the case for hazards coming from SoS communications. Emerging hazards, in turn, can be decomposed into *reconfiguration*, *interoperability* and *integration* hazards. Reconfiguration hazards might happen when switching from one state to another in the SoS. Interoperability hazards are emerging from possible misunderstanding between systems within the SoS, i.e., if a system command is interpreted in a way inconsistent with an intent of another system of SoS. Interoperability hazards are thus related to a SoS failure caused by a cooperation of correctly working systems. The last type, integration hazards, is caused by the integration of several systems into the SoS, e.g., by their interactions or resources sharing. Integration hazards are related to a SoS failure caused by its system failure within the integrated structure. This group has a subgroup related to *interface* hazards, i.e., hazards caused by a failure of at the input/output of a system.

The right side of Fig.2 depicts a part of FTA, i.e., a branch of an interface hazard. One of the faults leading to this hazard can be a failure of accessing a communicational channel between systems, as in this case a system cannot receive and transmit data at the correct time instances under the assumption of time-slotted access to the channel. A traditional FTA would have three single undesired events leading to this failure, namely a random hardware failure, a random software failure and a failure caused by the environment. For example, a sudden drop in temperature can put a node in an unsynchronized state, as temperature can influence the clock. However, for the considered use case, the harsh environment is mostly caused by dust. To include threats causing CSyn failure, we consider also an undetected malicious disruption of CSyn. A further decomposition of this failure is done by ATA, identifying all intentional attacks causing a CSyn disruption. We state that there is a branch in FTA leading from the CSyn failure to an SoS failure. We consider only a subtree of the CSyn failure to demonstrate its connection to ATA, since an SoS failure can potentially lead to a hazardous event. For example, a

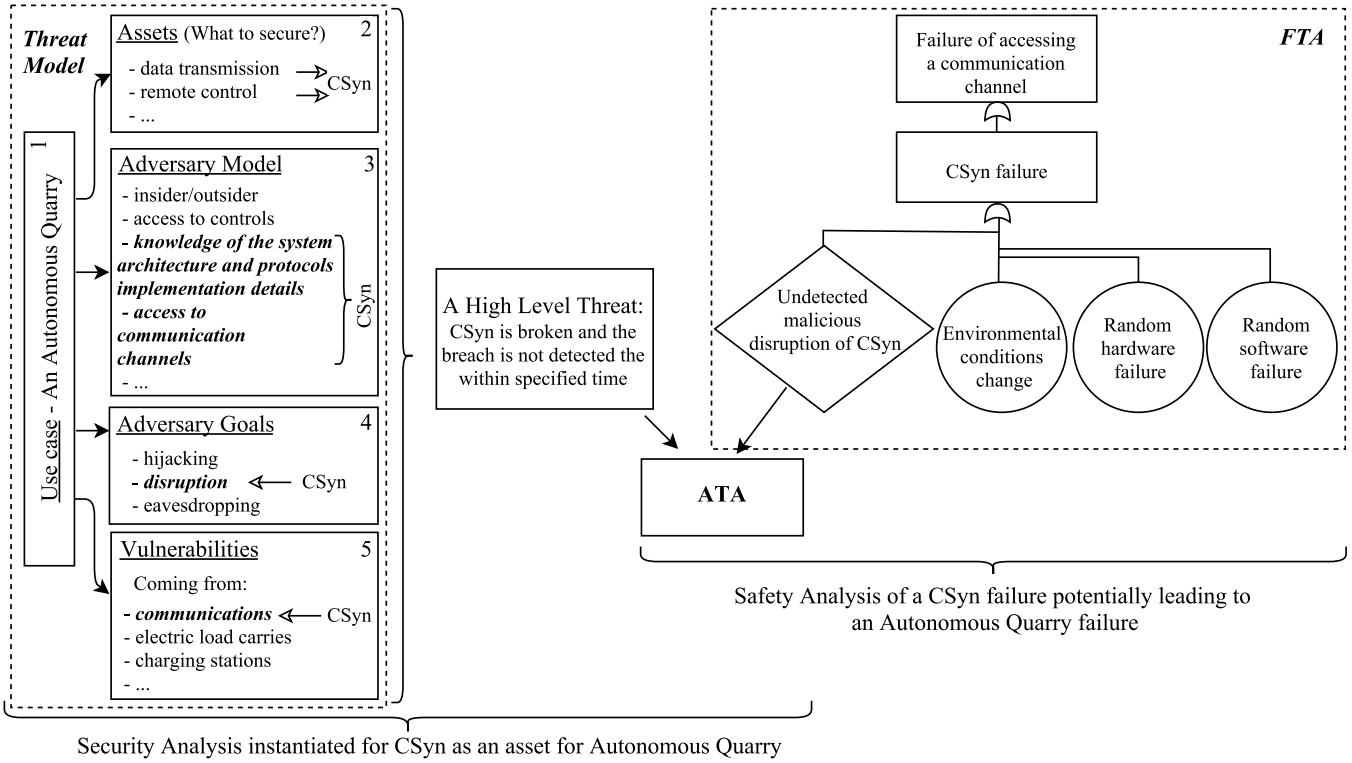

Fig. 2. Joint Safety and Security assurance approach for an Autonomous Quarry and its asset — a clock synchronization

collisions of electric load carries during their operation that might be caused by an SoS failure, namely their incorrect positioning due to disrupted communication with a control unit due to a CSyn breach, implies a hazardous event. Therefore, CSyn protection is relevant for safety assurance of the SoS. In general, Fig. 2 demonstrates how an outcome of the security analysis (the left hand side in Fig. 2), can be used as an input for the safety analysis (the right hand side in Fig. 2).

B. Joint Safety and Security Argumentation

Given the approach presented in Section V-A, we have identified communication as a potential vulnerability that could lead the whole system into an unsafe state. In Fig. 3 we depict an argument for a CSyn assurance case expressed with GSN.

The presented argument is an extension of the general argumentation for a safety assurance case presented by Troubitsyna [35]. The author follows the STAMP approach and considers three subgoals supporting the strategy of conducting STAMP-based analysis for safety assurance, that include: *(i)* assurance that the controlling software has a correct model of the controlled process; *(ii)* assurance that the logic of the controlling software is correct; and *(iii)* assurance that the controlling actions are implemented correctly. The first subgoal can be achieved via a strategy aiming to assure that malicious and accidental faults do not distort the model of the controlling process, considering safety and security together. In our work we follow the STPA-SEC approach and therefore we propose to add one more subgoal to this strategy in order to adjust the method to consider security caused failures, namely

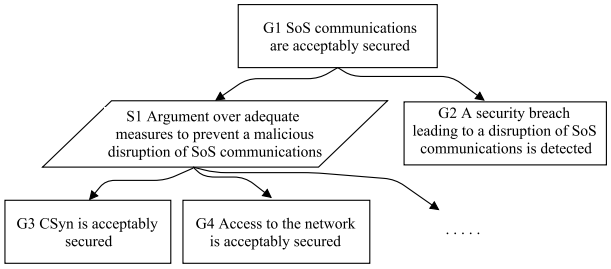

Fig. 3. A Clock Synchronization Argument

*G*1 — *SoS communications are acceptably secured.*

To assure this goal, a strategy and a goal are proposed, namely *S*1 — *an argument over adequate measures to prevent a malicious disruption of SoS communications*, and *G*2 — *a security breach leading to a disruption of SoS communications is detected*. The latter is formulated as a goal, as we do not consider its development further in this work, but it is needed in case the security attacks could not be prevented in order to make the SoS able to detect the existence of the breach and start switching into a safe state. In some cases, an attack needs time to its consequences become significant and if detected before, the harm of the attack can be minimized. For example, to have consequences, CSyn should be broken for some resynchronization intervals and the amount of intervals needed to disrupt the system depends on the system being under attack, e.g., how often it communicates. *S*1 can be realized by several subgoals, and it should be noted that we present only two of them: *G*3 relating CSyn and *G*4 relating access control. We do not expand it further as the purpose of

this work is to highlight the significance of communication safety and security assurance, as well as demonstrating how their overlap can be used to enrich the analysis and indicate crucial importance of considering CSyn with respect to communication security and SoS safety.

VI. Conclusions and Future Work

Given the increasing integration of a large number of computing and sensing devices and systems with different types of network infrastructure, security has become a serious issue while ensuring system safety in autonomous systems of systems, including safety-critical construction equipment. We can conclude that in order to guarantee safety in such systems, one has to include reasoning about security as well.

In this paper, we present our findings on exploring interdependencies between safety and security, focusing on threats towards breaching a clock synchronization and its possible effect on system safety in complex systems of systems. We propose an approach where we combine security and safety analyses into one process while addressing a common high-level goal, in our case protecting clock synchronization in an autonomous quarry with several cooperating vehicles. We go one step further and present joint safety and security clock synchronization argumentation using goal structuring notation, where we propose an extension by including communication aspect into the reasoning as it proves to be an important asset in complex autonomous systems of systems.

In future work, we aim at exploring other types of security breaches that affect systems safety in complex autonomous systems of systems and work towards proposing a general approach when jointly reasoning about these properties in systems of systems.

Acknowledgment

This work is funded through PiiA program and RAASS project, ECSEL JU SafeCOP project and by the Swedish Knowledge Foundation funded project SAFSEC-CPS. The SafeCOP project is funded from the ECSEL Joint Undertaking under grant agreement n692529, and from National funding.

References

[1] A. Burns, J. McDermid, and J. Dobson, "On the meaning of safety and security," *Comput. J.*, vol. 35, no. 1, 1992.

[2] D. F. C. Brewer, *Applying Security Techniques to Achieving Safety*. Springer London, 1993.

[3] J. Rushby, "Critical system properties: Survey and taxonomy," Computer Science Laboratory, SRI International, Tech. Rep. SRI-CSL-93-1, 1994.

[4] D. P. Eames and J. Moffett, *The Integration of Requirements*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.

[5] C. W. Axelrod, "Applying lessons from safety-critical systems to security-critical software," in *IEEE Systems, LISAT*, May 2011.

[6] T. Aven, "A unified framework for risk and vulnerability analysis covering both safety and security," *Reliability Engineering & System Safety*, vol. 92, no. 6, 2007.

[7] C. Ponsard, G. Dallons, and P. Massonet, *Goal-Oriented Co-Engineering of Security and Safety Requirements in Cyber-Physical Systems*. Springer International Publishing, 2016.

[8] T. Srivatanakul, J. A. Clark, and F. Polack, *Effective Security Requirements Analysis: HAZOP and Use Cases*. Springer Heidelberg, 2004.

[9] R. Winther, O.-A. Johnsen, and B. A. Gran, *Security Assessments of Safety Critical Systems Using HAZOPs*. Springer Heidelberg, 2001.

[10] W. Young and N. Leveson, "Systems thinking for safety and security," in *Proceedings of the 29th Annual Computer Security Applications Conference*. ACM, 2013.

[11] D. Schneider, E. Armengaud, and E. Schoitsch, *Towards Trust Assurance and Certification in Cyber-Physical Systems*. Springer, 2014.

[12] M. G. Doyle, "How Volvo CE is engineering a quarry run by electric loaders and haulers for big cuts to costs and emissions," http://www.equipmentworld.com/how-volvo-ce-is-engineering-a-quarry-run-by-electric-loaders-and-haulers-for-big-cus-to-costs-and-emissions/, 2016, [Online; accessed 18-April-2017].

[13] H. Kopetz and W. Ochsenreiter, "Clock synchronization in distributed real-time systems," *IEEE Transactions on Computers*, vol. C-36, no. 8, pp. 933–940, Aug 1987.

[14] M. Bouzeghoub, "A framework for analysis of data freshness," in *Proceedings of the 2004 International Workshop on Information Quality in Information Systems*, ser. IQIS '04. New York, NY, USA: ACM, 2004, pp. 59–67. [Online]. Available: http://doi.acm.org/10.1145/1012453.1012464

[15] R. Kissel, *Glossary of key information security terms*. U.S. Dept. of Commerce, National Institute of Standards and Technology, 2006.

[16] B. Schneier, "Attack trees," *Dr. Dobb's Journal*, vol. 24, no. 12, pp. 21–29, 1999.

[17] N. G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press, 2011.

[18] J. Fussell, *Fault Tree Analysis - Concepts and Techniques*. Noordhoff Publishing Co., 1976.

[19] D. Haasl, N. Roberts, W. Vesely, and F. Goldberg, *Fault tree handbook*, Jan 1981.

[20] T. Kelly, "Arguing Safety — A Systematic Approach to Managing Safety Cases," Ph.D. dissertation, University of York, York, UK, 1998.

[21] T. Kelly and R. Weaver, "The goal structuring notation–a safety argument notation," in *Proceedings of the dependable systems and networks 2004 workshop on assurance cases*. Citeseer, 2004.

[22] "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. 1–269, July 2008.

[23] F. Terraneo, L. Rinaldi, M. Maggio, A. V. Papadopoulos, and A. Leva, "Flopsync-2: Efficient monotonic clock synchronisation," in *2014 IEEE Real-Time Systems Symposium*, Dec 2014, pp. 11–20.

[24] D. Rylander, "Productivity improvements in construction site operations through lean thinking and wireless real-time control," December 2014.

[25] H. Hartenstein and K. P. Laberteaux, *VANET: Vehicular Applications and Inter-Networking Technologies*. Wiley, 2010.

[26] V. Chiprianov, L. Gallon, M. Munier, and V. Lalanne, "Challenges in security engineering of systems-of-systems."

[27] D. G. Lubas, "Department of defense system of systems reliability challenges," in *RAMS 2017*, Jan 2017, pp. 1–6.

[28] D. Mills, "Network time protocol (version 3) specification, implementation," 1992.

[29] "IEEE standard for local and metropolitan area networks– timing and synchronization for time-sensitive applications in bridged local area networks– corrigendum 1: Technical and editorial corrections," *IEEE Std 802.1AS-2011/Cor 1-2013 (Corrigendum to IEEE Std 802.1AS-2011)*, pp. 1–128, Sept 2013.

[30] B. Hirschler and A. Treytl, "Validation and verification of ieee 1588 annex k," in *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2011 International IEEE Symposium on*, Sept 2011, pp. 44–49.

[31] E. Lisova, E. Uhlemann, W. Steiner, J. Akerberg, and M. Bjorkman, "Risk evaluation for clock synchronization from arp poisoning attack in industrial applications," in *IEEE ICIT 2016*, 2016.

[32] E. Lisova, M. Gutiérrez, W. Steiner, E. Uhlemann, J. Akerberg, R. Dobrin, and M. Bjorkman, "Protecting clock synchronization: Adversary detection through network monitoring," *Journal of Electrical and Computer Engineering*, pp. 1–13, 2016.

[33] E. Lisova, E. Uhlemann, J. Akerberg, and M. Bjorkman, "Towards secure wireless TTEthernet for industrial process automation applications," in *Proceedings of the ETFA 2014*, Sept 2014, pp. 1–4.

[34] P. Redmond, 2007. [Online]. Available: https://calhoun.nps.edu/handle/10945/3679

[35] E. Troubitsyna, "An integrated approach to deriving safety and security requirements from safety cases," in *IEEE COMPSAC 2016*, vol. 2, June 2016, pp. 614–615.