Mälardalen University Press Licentiate Theses
No. 232

SECURING CLOCK SYNCHRONIZATION IN INDUSTRIAL HETEROGENEOUS NETWORKS

Elena Lisova

2016

MÄLARDALEN UNIVERSITY
SWEDEN

School of Innovation, Design and Engineering

ISBN 978-91-7485-258-5
ISSN 1651-9256
Printed by Arkitektkopia, Västerås, Sweden

Abstract

Today, wireless solutions for industrial networks are becoming more and more appealing since they increase flexibility and enable the use of additional wireless sensors, but also bring such advantages as mobility and weight reduction. Wired networks, on the other hand, are reliable and, more importantly, already existing in most distributed control loops. Heterogeneous networks consisting of wireless as well as wired sub-networks are gaining attention as they combine the advantages of both approaches. However, wireless communication links are more vulnerable to security breaches because of their broadcast nature. For this reason, industrial heterogeneous networks require a new type of security solutions, since they have different system assets and security objectives. This thesis aims to secure industrial heterogeneous networks. Such networks have real-time requirements due to interaction with some physical process, and thus have a schedule with one or more deadlines for data delivery in order to comply with the timing requirements of the application. The necessity to follow the schedule implies that all network participants should share the same notion of time and be synchronized. This fact makes clock synchronization a fundamental asset for industrial networks. The first step towards developing a security framework for industrial heterogeneous networks with real-time requirements is therefore to investigate ways of breaching clock synchronization. Once the vulnerabilities of this asset have been identified, the next step is to propose solutions to detect malicious attacks and mitigate their influence. The thesis provides a vulnerability analysis of the asset synchronization based on the widely deployed IEEE 1588 standard, and identifies a possibility to break clock synchronization through a combination of a man-in-the-middle attack and a delay attack. This attack is appealing to an adversary as it can target any network requiring synchronization. Next, several mitigation techniques, such as a relaxed synchronization condition mode, delay bounding and using knowledge of existing environmental conditions, are identified, making the network more resilient against these kinds of attacks. Finally, a network monitor aiming to detect anomalies introduced by the adversary performing attacks targeting clock synchronization is proposed as a mean to detect the delay attack.

Sammanfattning

Idag spelar trådlösa kommunikationsnätverk en allt viktigare roll inom industrin, eftersom dessa ger större flexibilitet och möjliggör komplettering med trådlösa sensorer i existerande tillämpningar. Dessutom ger trådlös teknik ett antal fördelar i form av ökad mobilitet och minskad vikt. De flesta existerande industriella nätverk är dock trådbundna, de är tillförlitliga men inte särskilt flexibla eller mobila. De mest lovande lösningarna, heterogena nätverk, blandar trådbundna och trådlösa nätverk. Detta kan kombinera fördelarna från de båda metoderna. Alla enheter i nätverket behöver kunna samarbeta och utföra komplicerade operationer, de måste kontrolleras för att uppnå korrekt prestanda, och de behöver förses med säkerhetsfunktioner, men eftersom trådlösa förbindelser kan höras och störas, krävs andra tillvägagångssätt för att tillhandahålla säkerhet.

I denna avhandling studeras säkerhet i industriella heterogena kommunikationsnätverk med realtidskrav. Realtidskrav innebär att meddelanden i systemet är känsliga för fördröjningar i tiden eftersom de interagerar med en fysisk process, och de flesta meddelanden har därför någon form av deadline. Många sådana system är dessutom säkerhetskritiska, vilket innebär att kostnaden för ett misslyckande är mycket hög. Därför behöver industriella realtidsnätverk schemalägga dataöverföringen med avseende på just deadlines. För att kunna följa schemat, måste alla nätverksdeltagarna vara överens om tiden – de måste ha synkroniserade klockor. Således är klocksynkronisering en av de grundläggande tillgångarna i ett sådant nätverk. Nuvarande standarder för klocksynkronisering tillhandahåller inte tillräckligt hög säkerhetsnivå. Som en konsekvens av detta är klocksynkronisering en svag punkt för många säkerhetskritiska realtidsnätverk. En säkerhetslösning som fokuserar på att skydda klocksynkroniseringen är därför en första nödvändig byggsten i ett ramverk för säkerhet i industriella heterogena nätverk.

Avhandlingen inleds därför med en undersökning av befintliga säkerhetsramverk för industriella nätverk, tillsammans med en utvärdering av deras användbarhet med avseende på realtidskrav och den heterogena karaktären av de tillhörande användningsfallen. För att förstå vilka mål som är rele-

vanta att säkra för ett visst användningsområde, föreslås en särskild metod för att modellera hoten mot nätverket. Klocksynkronisering utmärks då inte bara som en tillgång som är viktig för korrekt funktionalitet i systemet, utan även en tillgång som är gemensam för alla industriella nätverk, vilket gör den särskilt attraktiv för en säkerhetsattack. Därför undersöks olika metoder för att tillhandahålla och underhålla klocksynkronisering, med den vanligen förekommande IEEE 1588-standarden som exempel. Avhandlingen visar att klocksynkronisering som upprättats enligt denna standard lätt kan störas på grund av bristande säkerhetsstöd. Möjligheten att störa klocksynkroniseringen genom att utföra en tvåstegs-attack, en kombination av en man-in-the-middle-attack och en efterföljande selektiv fördröjningsattack, studeras i avhandlingen. Slutligen föreslås ett antal metoder för att minska konsekvenserna av en sådan attack, jämte ett sätt att skydda klocksynkroniseringen genom övervakning och trafikanalys av nätverket.

To my beloved grandfather
Моему любимому дедушке,
Лисову Василию Федоровичу

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors Dr. Elisabeth Uhlemann, Prof. Mats Björkman, Dr. Johan Åkerberg, and Dr. Wilfried Steiner. Thank you for guiding me, sharing your experiences and giving valuable comments and suggestions! You have always provided me with a great support and have encouraged me to jump higher and not to be afraid of falling. I learned a lot from you and hope we will continue working fruitfully together.

Furthermore, I would also like to express my gratitude to my supervisors at bachelor and master theses, Prof. Makarov Sergej Borisovich and Dr. Gelgor Aleksandr Leonidovich. You have inspired me to continue my studies as a PhD student.

My sincere gratitude goes to RetNet project members, our Tremendously Talented Team (TT–Team). I am very grateful to Hans Hansson, Mats Björkman, Elisabeth Uhlemann, Wilfried Steiner, Radu Dobrin, Sasikumar Punnekkat, Johan Åkerberg, Arjan Geven and Guillermo Rodriguez-Navas. I also would like to thank the project managers Caroline Blomberg and Christian Reinisch, Ann-Catrin Davis and Carolina Reyes for their support and help especially during secondments! And a special thank you goes to the students and my co-authors besides supervisors from this project (in the order of ESR numbers) Pablo, Marina, Francisco, and Ayhan – thank you for all those amazing trips and useful collaborations, now I know what Crazy Shark is, how to rock with guitar and the most important words in Spanish and Bulgarian!

I wish to express my appreciation to lecturers and professors at MDH from whom I learned a lot during courses, meetings and discussions. Thank you Gordana Dogid-Crnkovic, Jan Gustafsson, Barbara Galina, Hans Hansson, Thomas Nolte, Christina Seceleanu, Dag Nyström, Paul Pettersson, Damir Isovic, Radu Dobrin, Kristina Lundqvist, Maria Lindén, Moris Behnam, Luis Almeida, and Lucia Lo Bello. I would also like to thank all IDT administration staff and especially Carola Ryttersson, Susanne Fronnå and Sofia Jäderén.

A great thank you goes to friends and colleagues, that I have met in our department, for all the fun we had together during these years at conference and vacation trips, fika and barbeques, parties and cinema, badminton, tennis and the step-counter competition. It makes it so much easier to work when surrounded by such cheerful people always ready to share a joke or help in case of any problem, such as getting visa in time or moving to a new apartment. Special thanks go to my officemates Per, Sara, Gregory, and Branko who kindly tolerate my questions, discussions and the opened window. Moreover, I have never had a clue before what senseless word "sixers" can indeed mean, thank you for this – it made me feel like home here!

Last but not least, I would like to say endless thank you to my family, especially to my mother Irina, my grandfather Vasilij Fedorovich and my aunt Olga! I am so lucky to have you all, you have always supported me and I dare to do so much in my life only because I am assured in the dependable home ground!

This work was supported by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme.

Elena Lisova
Västerås, April 2016

List of Publications Included in the Licentiate Thesis[1]

Paper A: ***Towards secure wireless TTEthernet for industrial process automation applications***, Elena Lisova, Elisabeth Uhlemann, Johan Åkerberg, and Mats Björkman, in the *Proceedings of the 19th IEEE International Conference on Emerging Technologies and Factory Automation* (ETFA), Barcelona, Spain, September 2014.

Paper B: ***A Survey of security frameworks suitable for distributed control systems***, Elena Lisova, Elisabeth Uhlemann, Wilfried Steiner, Johan Åkerberg, and Mats Björkman, in the *Proceedings of the International Conference on Computing and Network Communications* (CoCoNet), Trivandrum, India, December 2015.

Paper C: ***Risk evaluation of an ARP poisoning attack on clock synchronization for industrial applications***, Elena Lisova, Elisabeth Uhlemann, Wilfried Steiner, Johan Åkerberg, and Mats Björkman, in the *Proceedings of the IEEE International Conference on Industrial Technology* (ICIT*)*, Taipei, Taiwan, March 2016.

Paper D: ***Protecting Clock Synchronization – Adversary Detection through Network Monitoring**,* Elena Lisova, Marina Gutiérrez, Wilfried Steiner, Elisabeth Uhlemann, Johan Åkerberg, Radu Dobrin, and Mats Björkman, to appear in the *Journal of Electrical and Computer Engineering, Hindawi*, vol. 2016.

[1] The included articles have been reformatted to comply with the licentiate thesis layout.

Paper E: ***Game theory applied to secure clock synchronization with IEEE 1588***, Elena Lisova, Elisabeth Uhlemann, Wilfried Steiner, Johan Åkerberg, and Mats Björkman, *MRTC technical report*, MDH-MRTC-309/2016-1-SE, Mälardalen Real-Time Research Center, Mälardelen University, March 2016 (a shortened version of this report is in submission to *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Stockholm, Sweden, September 2016).

Contents

I
Thesis

Chapter 1
Introduction

Today, communications play an important role in industry, as more and more functions are performed by geographically distributed special devices and equipment. These devices need to be connected, to cooperatively perform more complex operations, and controlled, in order to maintain appropriate performance levels while guaranteeing the required security and safety functions. In order to satisfy the growing complexity of industrial networks, wireless solutions are gaining more and more attention. Wireless technologies provide a number of obvious advantages such as increased flexibility, less weight, higher mobility, etc. [1]. On the other hand, as wireless connections by nature are broadcast networks, they require special approaches for providing an appropriate level of security functionality [2]. Wired networks, on the other hand, are inherently more reliable and, in addition, already existing in most distributed control loops. Many promising emerging solutions are therefore a mixture wired and wireless networks, so-called heterogeneous networks, which can take advantages from both approaches [3].

In this thesis, security in industrial heterogeneous networks with real-time requirements is considered. Applications with real-time requirements imply that messages in the system are time-sensitive due to interaction with some physical process. Therefore, to comply with the timing requirements of the application, the networks have a schedule for data transmission with one or more deadlines for data delivery. In addition, many such systems are safety-critical, which means that the cost of failure is very high. To be able to follow the schedule, and guarantee proper functionality of the system, the network participants must agree on a common notion of time. Thus, clock synchronization is a fundamental asset for the considered networks. There are different standards capable of providing clock synchronization in industrial networks, such as IEEE 1588 [4], but they do not provide the necessary level of security protection. As a consequence, clock synchronization can be considered a weak spot for most safety-critical industrial networks, since synchronization is an asset, which is common to many applications. Therefore, it is more likely that an adversary will invest recourses in its breaching, as one type of attack can be applied to several different types of applications. Consequently, a security solution protecting this asset is an important piece of the puzzle designing a security framework for industrial heterogeneous networks.

The overall scope of the thesis is to secure communications in industrial heterogeneous networks. To this end, this thesis presents an overview of existing security frameworks for industrial networks along with their evaluation and applicability considering both real-time requirements and the heterogeneous nature of the targeted use cases. In order to understand which security objectives that are relevant for a certain application area, an approach of modeling the threats is proposed. One part of this approach consists of defining a set of system assets. Clock synchronization is then identified as a common asset for most industrial networks. Considering the identified asset, an analysis of different ways for a malicious adversary to break it, an investigation of how to protect it by means of network traffic monitoring, and a formal analysis of the possible interactions between these two parties; the adversary and the network monitor, are provided. Firstly, different ways of clock synchronization provision and maintenance are investigated, using the widely adopted IEEE 1588 standard as an example. It is shown that due to the lack of security support, clock synchronization established according to this standard can be breached quite easily. Secondly, the thesis considers the possibility to break clock synchronization by performing an attack consisting of two phases: a combination of a man-in-the-middle (MIM) attack followed by a selective delay attack. Next, the thesis proposes to use network monitoring and traffic analysis as a way to protect clock synchronization. Finally, possible interactions between an adversary and the monitoring system is investigated and formally analyzed by applying game theory.

1.1. Problem Formulation

In order to meet emerging market demands industrial networks require a security solution that can cope with increasing network heterogeneity. From a security point of view, the heterogeneity implies that a reevaluation of the system assets and the required security objectives is needed. *The goal of this research is to define the most critical assets of industrial heterogeneous networks, and to develop appropriate security mechanisms to protect them.* In the development process, the requirements for real-time support and safety-critical applications are considered. The next step is to integrate the developed solutions into one complete security framework, covering all relevant assets. As there are several critical assets for industrial networks, and as some security solutions may impede the real-time requirements by introducing additional overhead an iterative approach is adopted.

One of the common characteristics of industrial networks is the joint requirements on reliability and timeliness Even for non-safety-critical applications, there are still deadlines for messages transactions and if a message misses its deadline, the functionality is lost or loses in quality. The necessity to meet deadlines implies that all network participants should share the same notion of time and be synchronized. Clock synchronization is thereby an essential asset for industrial networks. In most standards used for industrial networks, the clock synchronization procedure does not have any security protection. Protecting clock synchronization can therefore be considered as an important step on the way to securing industrial heterogeneous networks.

Given the goal and the problem described above, the following hypothesis is used in the thesis:

Hypothesis: it is possible to develop a security framework for heterogeneous industrial networks, where the adopted security solutions are selected and designed according to the specific requirements from safety-critical real-time applications.

To reach the goal, the following research questions must be addressed:

Research questions

- RQ1: What are the main system assets and security objectives for heterogeneous industrial networks?
- RQ2: How can the main system assets be protected and the needed security objectives provided in the presence of real-time requirements?
- RQ3: How should the possible interactions between an adversary and the network be analyzed while taking into account learning ability of both sides?

1.2. Research Method

When conducting research in a specific area, it is important to understand the related research methods and be able to apply them. A general framework for research methods in the computing area is presented in [5]. It consists of a repetitive circle with four main steps: specifying what to be achieved; specifying where the data can be collected and how it can be used; choosing the way to process the data; and finally evaluating the results and analyzing whether the goals have been achieved or not. For the research work conducted in this thesis, this circle can be adjusted according to Fig. 1.1. The research work starts with the *problem formulation*: defining the goal of what is to be achieved, specifying the questions that are needed to be answered on the way and relating the research problem to state-of-the-art in the area. The next step is *solution development*, where, based on existing techniques and appropriate analysis of these, a proposal is developed regarding how to solve the problem or a part of it. Usually, a solution can be presented as a combination of small parts that, assembled together, build the required framework. Once a solution is proposed, it should be evaluated in the step *solution evaluation*, using related performance metrics. Finally, the last step in the circle is *results evaluation*. This step is needed to analyze if what has been derived is what was actually wanted and formulated in the goal of step one. As an outcome, possible refinements can be identified and applied to the goal, after which the circle can be repeated until the results are satisfying and the goal has been achieved.

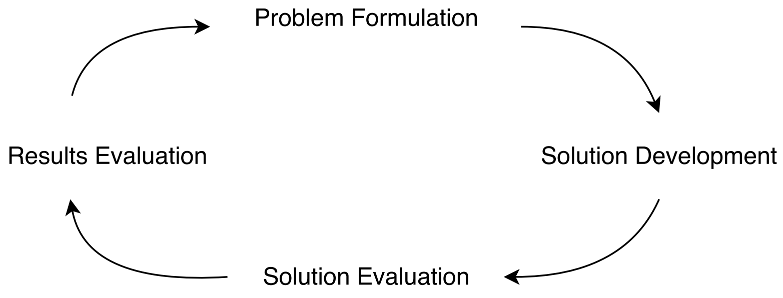

Fig. 1.1 Research Circle.

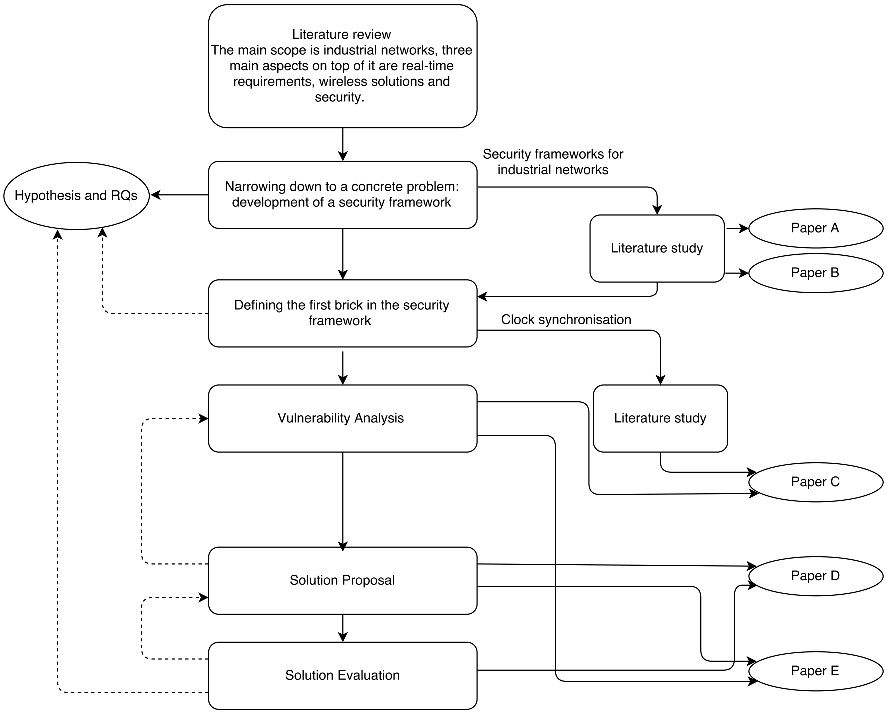
Literature review
The main scope is industrial networks, three main aspects on top of it are real-time requirements, wireless solutions and security.
Hypothesis and RQs
Narrowing down to a concrete problem: development of a security framework
Security frameworks for industrial networks
Literature study
Paper A
Paper B
Defining the first brick in the security framework
Clock synchronisation
Vulnerability Analysis
Literature study
Paper C
Solution Proposal
Paper D
Solution Evaluation
Paper E

Fig. 1.2 Research process.

The research methodology adopted in this thesis use these four steps, which also includes a literature reviews and study, collaboration with industrial partners in order to relate the research to state of the practice, using the tools AVISPA and OMNeT++ to conduct attack and solution evaluations with and, finally, presenting and disseminating the results by publishing papers. A more detailed representation of the research methodology is shown in Fig. 1.2, where the main steps of the adopted research process are presented with rectangles, the main results are depicted with ellipses and the arrows demonstrate the flow of the process, with dashed arrows implying feedback loops that involves iterations in the process with small refinements at each stage.

1.3. Thesis Outline

This thesis is organized into two parts, a comprehensive summary and a set of appended research papers. The reminder of the first part has the following structure: Chapter 2 describes industrial heterogeneous networks, and their corresponding application requirements, while Chapter 3 outlines the adversary goal and system assets. Next, Chapter 4 details the security objectives when securing industrial heterogeneous networks, including ethical aspects of security. Finally, Chapter 5 outlines the thesis contributions and highlights their importance. In addition, Chapter 5 also includes a brief description of the appended research papers together with an explanation of the authors contributions in each of them. Finally, Chapter 6, containing a summary and directions of future research work, concludes the first part. The second part is a collection of five papers, on which the thesis is based.

Chapter 2
Industrial Networks

In this chapter, the networks constituting the main focus of the thesis, namely industrial networks, are described. One of the most important issues for providing secure communications in industrial heterogeneous networks is to derive and formulate the related requirements that should be supported by the proposed security solutions. Four main requirements are considered: reliability, timeliness, availability and heterogeneity. This implies that the proposed security solutions not only need to ensure secure communications, but also that security is not achieved at the expense of any of the other main application requirements.

2.1. Application Requirements

Industrial applications include such examples as airplanes, spaceships, robots, production lines at factories and so on. More and more such applications require safety and security as provided services. To be able to provide such services, the system should support a set of requirements. The following four are the most main ones [6], although of course, considering a concrete use case, other relevant requirements such as energy efficiency or low delay can be added.

High reliability. If a system is reliable, it means it has a low probability of errors and consequently a low probability of failure such that there are mechanisms embedded to handle the failures and mitigate the consequences if preventions and elimination techniques should fail. The question is how "low" the failure probability should be in order for the system to be considered reliable enough. For a production line, this will be connected with the potential money loss implied by an unwanted stop in production. For example, with a paper machine, there is a huge economic loss even for minutes of standstill. Hence, the cost of idling and the cost of developing mechanism for increased reliability to protect against idling will be compared. In examples involving possible harm to humans, fault-tolerant techniques are used to make failure probabilities as small as possible, even at the expense of functionality. From a security point of view, any proposed security solution should not influence the reliability of the system or at least this influence should be estimated and checked for acceptance.

Timeliness. By timeliness, the ability to cope with real-time requirements is understood. It means that deadlines associated with information, messages, or services provision shall be respected, not only in terms of not being too late, but also not being too early. Nowadays, the vast majority of industrial systems have some kind of real-time requirements. Using an airplane as an example, its landing gear should unfold within the allowed time range, i.e., not in mid-air, before the landing procedure has been initiated, and not several minutes after the plane has landed. From a security point of view, this requirement is very challenging as almost all security solutions introduce communication overhead in the system which implies increased delays and causing tighter timing requirements.

Availability. By availability it is meant that the system should be able to provide the required services without a failure or, in some cases that it should be able to withstand a failure due to a malicious or harmful environment. It is a core requirement for safety-critical services, where the cost of such failures is high. In the case of an electric power steering (EPS) system in a car, the assistance should be provided even if there is failure of the battery, or, at least, techniques decreasing the severity of the consequences in case of failure should be implemented. Whenever a security solution is added on top of a system architecture, it is complicated to analyze all possible interconnections and evaluate its influence on availability. Therefore, it is important to consider the security requirements and their interconnections with system requirements already at the stage of system development.

Heterogeneity. This is not a traditional requirement, but it is added here to broaden the targeted application area, according to today's market demands. There is a number of advantages with wireless solutions, and thus more and more technologies are looking towards wireless extensions, possibly as a mixture of wired and wireless networks. However, to be applied in industrial networks, the wireless solutions should be able to provide services with at least the same quality as purely wired ones. From this perspective, security is a potential showstopper as it is difficult to provide security solutions with heterogeneity support. Wired solutions are often already existing, and thus when adding wireless solutions, complementing security solutions has to be added on top of an existing system architecture. In addition, wireless links has an open nature, and therefore they require specific security solutions that can address issues such as broadcast and interference, while simultaneously comply with the rest of the requirements. It should also be noted that the term heterogeneity actually has a broader meaning than simply the mix of wired and wireless networks, as discussed below.

2.2. Heterogeneous Networks

Today, several meanings of the term heterogeneity can be found in the literature [7, 8]. As mentioned above, a mixture of wired and wireless connection links implies two types of link layers, medium access control (MAC) methods and physical layers coexisting in one network. Besides this, the end nodes can be different, meaning that they can consist of different hardware and, thus, have different memory, power supply, computational capacity etc. Such systems need different approaches for different sub-networks, or alternatively, the whole network must satisfy the most rigorous requirements of any one specific node, e.g., adjusting to the node with the smallest memory footprint or the lowest computational capacity, or the lowest transfer rate.

Heterogeneity can also be considered on another abstraction level, namely the data level where different types of data traffic exist in the networks. Referring to e.g., TTEthernet technology [9], the following data types can be considered: Best-Effort (BE), Rate-Constrained (RC) and Time-Triggered (TT). BE traffic does not imply timing constraints, and therefore, there are usually no guarantees on the arrival time of the messages. TT traffic has the opposite nature. It is periodic and requires low jitter. Hence, this type of traffic is usually used in safety-critical applications, where the cost of failure is higher than the cost of inefficient use of the bandwidth. RC traffic is somewhere in between the two previously described traffic classes. It has an event-driven nature and allocated scheduled time-slots, but at the same time, it has constraints in terms of the maximum consumed bandwidth within a given time-period. In this research, the target is do design a security system that is as useful as possible. Therefore, all three considered types of traffic classes are possible and taken into account.

Chapter 3
Adversary Goals and System Assets

As security risks are becoming a showstopper for deployment of industrial heterogeneous networks, deployment of appropriate security solutions is considered an increasingly important requirement [10]. Security risks exist if there is a vulnerability, such as a design flaw or some weaknesses in terms of oversimplified passwords or keys [2] and a threat, i.e., some value or advantage that may be gained from breaching security. The term "security" covers a wide range of provided services, so-called security objectives, ranging from error control codes to compromised node detection. A security objective describes what type of threat the system needs to be secured against. Different applications have different security objectives. In some networks, data is not confidential and all that is needed is data origin checking, whereas in other networks it is crucially important to keep data confidential, due to e.g., the need to protect product recipes.

In order to reach the security objectives and provide adequate levels of security service, potential adversary goals, i.e., desired targets for an attack, should be investigated, as this allows to predict the adversary actions and estimate the possible consequences of an attack. The system assets and the adversary goals can be derived from the considered use case [11]. In this chapter, some typical adversary goals regarding industrial networks are identified and discussed. Furthermore, a possible set of relevant assets for industrial networks, is highlighted, with special focus on clock synchronization, as this is a common denominator for many distributed control applications. Adversary goals and systems assets are like two sides of the same coin, and even though they do not always coincide completely, they do reflect each other and they are interconnected, since an adversary is more likely to target the most valuable properties in the system.

3.1. Adversary goals

There are many possible goals for an adversary targeting industrial heterogeneous networks, but according to [12] the following three can be considered as general adversary goals: system disruption, eavesdropping and system hijacking. All three are possible within the considered use case industrial applications and may lead to that the adversary can control or interfere with something from a distance if security is breached. If we consider a power plant generating electric power to a city, the adversary can try to disrupt the system. For factories producing products according to secret recipes, the adversary is likely to target system eavesdropping. Finally, an adversary can target automated factory hijacking or vehicle control hijacking which can lead to significant damages or even loss of human lives.

The three main adversary goals disruption, eavesdropping and hijacking are briefly described below, along with examples from industrial applications for which they are relevant.

Disruption. This is a quite broad term, as it includes any unplanned or unspecified behavior or malfunctioning of the network, ranging from a delay in service provision till complete shutdown of the system. The longer the disruption stays undetected, the bigger the consequences usually are. However, in this case the adversary does not control the network, but simply affects the network availability. A denial of Service (DoS) attack is a classic example of disruption, resulting in that the functions of the network are paralyzed even if though the adversary is unable to access information in the network or resetting its settings.

Eavesdropping. This target indicates that an adversary is interested in information transmitted or circulating in the network. It is a prime goal for networks containing confidential information, like production procedures or secret recipes. This adversary goal closely relates to two security objectives, namely, confidentiality and privacy. A common countermeasure is cryptography, which can limit the access to confidential information.

Hijacking. If an adversary manages to hijack the system, it means that the system behavior can be adapted and changed from the outside. Basically, it implies that an adversary takes control over the network or the related system. If the system is the brakes in a car, it means that the driver no longer can influence the brake. The specific characteristic here is that the intrusion may not be detected until the provided services differs, since the adversary penetrates the network and masks its actions as being benign until the gain of system interference is maximized.

The three adversary goals described above cover most of the threats industrial networks are subject to, but of course depending on the use case there can be others. For instance, forging, when the adversary wants to replace the information in the network with disinformation. In the example with a car in an intelligent transport system the driver being the adversary can consciously change, i.e., forge, saved data about the position of the car, to avoid road tolls.

3.2. Assets for Industrial Networks

System assets define what is needed to be protected in the system. These can roughly be separated into two groups: assets general for the area of industrial networks and assets that are application specific. Note also that there is a distinction between system requirements and security objectives, and the difference is that assets are real physical or intellectual properties of a system, whereas requirements relates to the expectations on the system performance.

Regarding application specific assets, there are several examples. In the EPS system mentioned above, the precision of the sensor measuring to quantify the characteristic of the driver actions can be an asset. It entails that sensor degradation by natural or malicious conditions should be prevented. In a paper machine with a sensor network measuring product humidity, data integrity can be considered as an asset, as the only requirement is the validity of the humidity values, since they are not confidential or secret, but need to be up to date and correct. In the system controlling pressure inside an airplane, bounded maximum latency in communication can be considered an asset, since it is critical to detect sudden reductions in cabin pressure as fast as possible.

Communication channel robustness can be considered as an asset common for all industrial networks. In case of wired communication links, it mainly means physical protection of wires and/or access to them. It is more complicated when it comes to wireless networks, as there, anyone with an antenna can receive the signal and try to process it. In this case, prevention techniques such as frequency hopping or masking the transmitted signal with noise, can be used.

As described above, support for real-time requirements is one of the main characteristics of industrial networks. To be able to meet deadlines and provide services in time, nodes in the network should share the same notion of time, i.e. by being synchronized. This is an essential asset for industrial networks because if synchronization is breached, it is possible to cause complete system disruption. There are many ways of influencing the synchronization, but one of the most difficult to detect and counteract is the delaying attack, as the adversary does not need to alter any data, but only to delay a few synchronization messages to put a node into unsynchronized mode [13]. Hence, clock synchronization is the main scope of this thesis and securing it is a primary goal to be achieved on the way to building a complete security framework for industrial networks.

3.3. Clock Synchronization

To be able to perform coordinated actions, the participants in industrial networks must share the same notion of time. Usually, each node has its own internal clock. In ideal situations, when all nodes have perfect clocks, this is enough for ensuring network reliability. In reality, however, clock precision depends on the cost of the clock – the more expensive the clock is, the more accurate it is. This is due to the fact that each clock has its own drift, i.e. the difference between the global time and its own internal time. If two communicating nodes have a clock drift that, relative to each other, is more than what is allowed in the system for correct functionality, this can lead to system failure.

In order to prevent possible hazards, clock synchronization algorithms are used to assist the nodes with frequent clock corrections. As the drift is a natural property of all clocks, clock drifts cannot be prevented, only mitigated by the periodic corrections. Clock synchronization algorithms aim at providing nodes with the values needed for periodic corrections. Only relative clock drift is important for considered types of applications. Note that it is often only the relative clock drift that is important for correct functionality in the considered applications. Therefore, clocks should be synchronized to each other rather than to the global time.

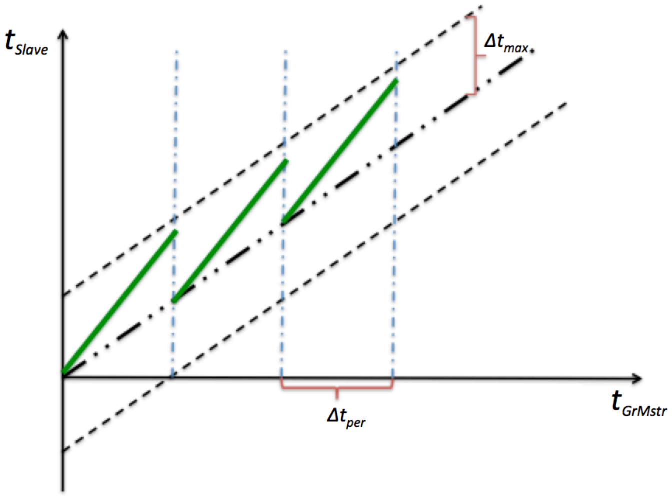

Fig. 3.1 Periodic correction of the clock drifts

The IEEE 1588 standard [4] is often used in industrial networks and distributed real-time systems for clock synchronization [14]. It has a master-slave approach, where there are one or more grandmasters with excellent clocks in the network and all others, the slaves, are correcting their clocks periodically to stay synchronized with a particular grandmaster. The offset needed for the correction is calculated via the exchange of so-called synchronization messages. The clock correction is typically periodic as shown in Fig. 3.1, so that the relative offset does not exceed the acceptable boundaries. The ideal relation between a slave clock time t_{Slave} and a grandmaster clock time t_{GrMstr} is shown in Fig. 3.1 with a dash-dotted line, whereas the potential real relation is shown by the green lines. Without corrections made every Δt_{per}, it can easily breach the boundaries for correct functionality (shown as dashed lines in the figure). IEEE 1588 also includes an algorithm to assign a new grandmaster if the current one fails. This causes the network to self-organize and become more robust. Even though IEEE 1588 has some recommended security services, such as message integrity and group authentication [4], there are still possibilities to disrupt the protocol, e.g. by conducting a delay attack [15].

Chapter 4
Security Objectives

A discussion about safety and security interconnections together with ethical issues concerning security are opening this chapter. Next, an overview of security approaches applicable to industrial networks is presented, and the main challenges and current state of the art in providing security services in the area are specified. Existing security solutions for wired as well as wireless networks are reviewed while highlighting their main differences and limitations.

4.1. Security and Safety

Although the main scope of this research work is security, safety is also involved as an additional requirement for the targeted application area. Therefore, it is important to define both terms, as well as how they are to be interpreted in this thesis. Traditionally, the two areas have been studied separately [16] and consequently, there are now different taxonomies, approaches and techniques used in each area. However, with the system complexity level increasing every day, safety and security are becoming more interconnected and vital for critical applications and, hence it is important to understand their similarities and differences.

Within dependability theory, an attribute can be considered as a system property through which system dependability can be reached. The work of Avizienis *et al.* [17] presents safety as one of the attributes of dependability and security as a composition of three attributes, namely Confidentiality, Integrity and Availability (CIA). In addition, the authors identify a set of secondary attributes to security, such as accountability, authenticity and non-reputability. From this perspective, security and safety can be regarded as completely different terms with no overlap. Kunze *et al.* [18] stress the difference between them by connecting safety to “protected systems” and security to systems “free of danger”. By the first definition is meant to consider and to set valid prevention and mitigation techniques, while the second definition is more vague as no definition of what can be understood as a danger is given. Further, the authors reflect on the difference between safety-critical and security-critical requirements. Safety-critical requirements target hazard illumination, which can be identified by conducting different types of safety analyses, e.g. Fault Three Analysis (FTA), Hazard and Operability Study (HAZOP), etc. For security, this process is more complicated as it is close to impossible to predict all possible actions and capabilities of a malicious adversary. However, one of the main differences between safety and security is the presence of malicious intensions in the case of security. Another distinction is the character of occurrence of events. For the safety case, the system is usually protected against some pre-specified number of consecutive events, and since there is no malicious intension, errors caused by humans will cease when the problem is detected. However, in case of security, a malicious adversary will not stop when detected, and instead the failure will continue until the adversary is isolated from the network, and even when this happens, the adversary may try to breach the system again. Working with safety requirements, hazards are identified, while working with security requirements, threats are usually considered. Within different application areas, there are related threat taxonomies, e.g., for internet infrastructures [19], embedded systems [20], wireless networks [21], service centric systems [22], etc. Within the different application areas, possible threats and attacks are grouped and classified differently, taking into account specific area requirements, e.g., the open nature of wireless communication links.

The conclusion is that although safety and security may be different areas associated with different approaches and methods, one cannot be assured without the other. A car can comply with all relevant safety standards, and thereby be considered safe enough, but if the system can be hijacked easily such that an adversary can take over the control, it is still not safe. Similarly, a production line, where communication is considered to be secured, it is still not good enough if the system cannot cope with non-malicious power failures and cases where a loss of power switches the equipment into a random mode. Similar examples raise the question about merged requirements for both safety and security [18]. There are attempts to integrate requirements from both areas, e.g., within embedded systems [23]. Further interconnections between safety and security and how one of them can be used or partly re-used to achieve the other one, already at the system design phase are investigated in [24]. The authors use the example of a system modeling language, and how the outcome of a safety analysis, e.g., using IF-FMEA, can be reused for security, e.g., STRIDE, and vice versa. This approach shows that the first step in working with both terms jointly can be taken through using the same language to describe them.

The approach adopted in this thesis for working with security while jointly considering safety is detailed in Paper A, but already here it worth-while mentioning that by safety is understood here what is broadly specified in [17] and by security a combination of the definition provided in [17] and the one provided in [25] are understood. In other words, by safety is understood reducing the probability of hazardous events to an accepted level and by security is understood a combination of services providing the required security objectives in presence of malicious intentions. Although the main focus of this thesis is security, the overall long-term goal is to bring both security and safety to the level required in mission critical heterogeneous network. As follows from the discussion above, both issues should be considered jointly and should complement each other in the developed security framework.

4.2. Ethical Aspects in Security

Nowadays, technologies are developing with tremendous speed, but they are still only tools in the hands of humans. Thus, how these tools are used depends on formal regulations, laws, and informal regulations, ethics. Ethics relates to well-founded and reasonable standards of what is right and what is wrong. It is less strict than a formal law, but still something bigger and more organized than the conscience. Ethics forms society and shapes the way it develops. Ethics also refers to the term of having rights. This is particularly important from a security point of view. The classical scenario considers a confrontation between an adversary on the one hand, and a network/security engineer/company owner on the other. However, it is not always clear who is the “bad guy”, as shown in e.g. [26] where the feasibility of information security is investigated. The authors therefore propose to extend the existing Information Systems Secure Interconnection (ISSI) model to consider five additional non-technical layers on top of the OSI model, with the highest one being ethics. In order to take security into consideration, the authors add group establishment rules and one more social contract layer. Such an approach allows to consider group interaction and human behavior as aspects of security. No direct interactions with humans are considered in this research work, apart from representing an adversary as an actual person. In addition, the thesis considers industrial information security rather than private personal data. Therefore, the ethical issues are not that pronounces in this thesis. However, it should still be considered to complete the picture, as capabilities to make ethical decisions may be needed in other safety-critical heterogonous networks, where the solutions developed in this thesis could be applied.

In this research work, an adversary is considered as an abstract party trying to influence the system. It does not matter whether the adversary is specifically related to a human or a specified device, and therefore the adversary can be referred to as “it”. Obviously, it is important to stay neutral and that if, during a discussion, human presence is assumed behind the malicious actions, the gender is not specified by addressing the adversary as “he” or “she”. Further, a simplified model of authorization is used in this thesis, where it is assumed that the communication information between benign network participants is authorized, belongs to the network and is not harmful to anyone. Consequently, it is implied that it is not unethical for an authorized user to monitor the data traffic in the network for security reasons. It is further assumed that the network availability is ethically correct, and thus only the adversary can be considered unethical. Also, proactive techniques are currently not considered, and thus there is no action from the network against adversary that necessitates considering ethical aspects.

In a more complicated ethical model, questions about who owns the data should be investigated. Consider the case with intelligent transport systems. It is increasingly difficult to predict the possible outcomes of an analysis of all collected data in vehicular networks. Knowing the daily route of a person, it is possible to predict many things, such as where the person works, buys grocery, sees friends, whether the person is sick, went to a hospital, etc. Thus, it is a complex task to predict the level of information significance and therefore assign appropriate protection techniques. In the future, the security framework developed here is planned to be extended, so that it can provide more diverse security services, and in this case ethics aspects regarding who has the rights to collected data needs to be reinvestigated.

Another ethical aspect in security is the delegation of responsibility in case of a failure. Keeping in mind the malicious intensions of an adversary, it is easier to determine the cause of a security failure than when failing to provide the required safety services [27]. Even though the prime source of responsibility is evident, it is challenging to claim a required quality level of provided service, as this claim must be verified and, in case of a breach, part of the responsibility also falls on the security service provider. Furthermore, even if the cause of failure is identified and the responsibilities are properly delegated, the question is still how malicious actions can be separated from non-malicious accidental errors. These two types of possible causes, of course, imply different reactions. However, this challenge is left for future work, as by now no level of clock synchronization protection is claimed and the investigation regarding a secure enough solution that can comply with industrial requirements is still ongoing. Consequently, there is an awareness about the ethical issues connected with the security in this research work, but the compete evaluation from an ethical point of view can only be done when the framework completed and the full set of solutions incorporated, and thus it is left for future work.

4.3. Security in Industrial Networks

Security is a hot topic for industrial networks and there are many research papers providing general descriptions and classifications of attacks, security countermeasures and challenges, e.g. [2, 12, 28]. Requirements for security in industrial networks along with an overview of solutions such as standards, intrusion detection systems, and countermeasures using cryptography, are presented in [29]. Security solutions can also be optimized to target the requirements of a specific application area. One of such areas is smart grids. A smart grid is a power network, using communication technology to intelligently integrate its users (power producers and consumers) and manage both electrical and informational connections. An overview of security challenges in this specific area, such as scalability and regulation issues, is presented in [30]. A security framework for micro-grids targeting communications between control elements was proposed in [31]. Here, the authors consider a triad of security objectives and constraints namely real-time performance, broadcast and multicast support. Each separate type of attack needs to be investigated in order to understand its possible consequences and develop countermeasures. However, current trends show that it is close to impossible to protect a system by aiming to eliminate all possible attacks, as there will always new ones coming. The protection system should instead be able to react to unknown malicious actions and be able to provide countermeasures against them. However, this approach has not been implemented in practice yet, and so far, even highly adaptive systems have a set of known attacks and related countermeasures, so that it can "educate" itself and learn how to adapt.

One type of threat for industrial networks is Advanced Persistent Threats that targets one particular goal and tries to reach it using several different approaches. One way to protect against such types of attacks is described in [32], where the authors propose a security solution following the "defense in depth" approach and taking as a basis the principle of immune "self-educating" systems. Possible countermeasures against Sybil Attacks, when an adversary masquerades as a legitimate network participant, such as trusted certification, recourse testing, or location verification, are investigated in [33, 34]. Often, industrial networks may contain sensitive and/or confidential information related to specific conditions, requirements of factory processes, or control information of network participants. In all these cases, the networks need to have adequate security properties to protect against malicious influence. Historically, the architectures of many wired industrial networks do not include any security solutions, as the networks are considered closed and static [35]. Especially, this is true for networks in environments that are physically hard to access, e.g. the embedded systems in cars. During the development phase of such systems, it is usually assumed that an adversary cannot have any access to the network. However, the amount of effort and resources that the adversary is willing to invest into system breaching is proportional to the possible gain of achieving the adversary goals. The focus in wired industrial solutions is essentially on safety requirements, as many applications are safety-critical, e.g. aircraft engine control system [36]. In such networks, a failure can cause harm to humans, the environment or result in a significant financial loss. However, wireless solutions are becoming more common in industrial settings. In these cases, there is an understanding of the risk of carrying sensitive information from the very beginning. Therefore, security services are usually considered already at the development phase of the systems. At the same time, safety requirements are usually handled in a more lightweight way, as, traditionally, wireless solutions are considered unreliable to begin with and thus not suitable for safety-critical applications. In the future, it is likely that merged approaches combining wired and wireless networks will bring both safety and security to appropriate levels. Emerging heterogeneous networks will need a security framework that is suitable for both wired and wireless connections. Such a framework should be flexible and provide different sets of security services for different parts of the network, based on its physical realization and the specific application area. Therefore, a suitable security framework should entail a set of interconnected solutions covering different security objectives. Such a structure allows the framework to adapt to particular use cases.

There are several challenges for industrial heterogeneous networks, such as support for different traffic classes, bringing together wireless and wired solutions, and providing relevant safety and security concepts and mechanisms [37]. When working with security, it is important to define the main characteristics of the required solutions, as security is a very broad term that has many interpretations. According to [17] security can be defined as a composition of its attributes or objectives, where the classical triad of objectives is CIA. The basic set of objectives can be completed with objectives related to specific use cases [2], such as authentication, authorization, auditability, etc. Today, industrial networks have a high level of complexity, thus applying security concepts in this area imposes several challenges. One of the main challenges relates to the significant number of interdependencies in such networks [38]. This means that usually security cannot be implemented efficiently on the top of a complete, existing architecture. There are several reasons for this, for example, that security solutions imply delays and communication overhead, factors which can be crucial for systems with high requirements on throughput or communication latency. To be efficient, such solutions should be considered already at the development phase and incorporated into the original system design.

Industrial networks have a number of specific differences compared to networks of desktops or office computers [10], such as longer expected lifetime, low tolerability to delays and outage, high reliability requirements, very rare network software upgrades, etc. Moreover, possible consequences of industrial system failures are quite severe. All these requirements together lead to the need for appropriate security solutions and unfortunately, existing standard solutions from information security usually fail to fulfill these specific requirements [11].

Among all types of industrial networks, the most challenging from a security point of view, given their functionality, is Industrial Control Networks (ICNs). These systems are responsible for decision making and switching between system states, and thus they are more sensible for malicious attacks and, simultaneously, have higher requirements on their functionality. ICNs have specific characteristics and requirements for security services. ICNs can be divided into the following categories [39]: Programmable Logic Controllers (PLCs), Supervisory Control and Data Acquisition (SCADA) and Distributed Control Systems (DCSs). PLCs are specialized devices consisting of a processor, interfaces, communication sub-modules, and a power supply. SCADA is responsible for data management and interaction with users. DCSs are used to control industrial processes such as water treatment, electrical power generation, etc. The main difference compared to SCADA is the process driven structure. DCSs are typically heterogeneous systems with real-time requirements [40], and this class of systems interact with both software and hardware and can fully use the advantage of mixed networks.

4.3.1. Protocols and Standards for Wired Industrial Networks

Industrial applications traditionally use wired networks. Therefore, there are well-known and established protocols and technologies available for wired networks. Two widely used application layer protocols are Modbus and Distributed Network Protocol (DNP3). The Modbus protocol was developed in 1979 by Modicon [41]. Later, it was extended to run over TCP networks, and this version of the protocol, called Modbus TCP, is a popular protocol as it is an open standard, and easy to deploy and maintain. Modbus uses a master-slave approach, where only the master can initiate communication, and slaves respond or take prescribed actions. Modbus was developed without considering security aspects. DNP3 is mostly used for SCADA, it aims to establish reliable communication between a SCADA master and remote servers [42]. DNP3 can also run over TCP and the protocol reliability is based on additional CRC checks. This mechanism can help against channel errors but not against malicious adversaries, and thus DNP3 is not secure.

Some examples from the automotive domain are the Controller Area Network (CAN) and Time-Triggered Ethernet (TTEthernet). The first one is broadly used for in-vehicle embedded systems. CAN is implemented on a bus, where all nodes can hear each other and there is no possibility to send a message to a specific node, as the nature of the protocol is broadcasting. From a security point of view, the only service provided with CAN is the possibility to detect whether a message was corrupted during its transmission. TTEthernet [43] is a platform that complements Ethernet so that it can be used for safety-critical applications with real-time requirements. The application area of this protocol is quite large and includes not only automotive, but also aerospace, space, railway, energy etc. However, security aspects are not considered here either.

Another protocol often used in industry is Highway Addressable Remote Transducer (HART) [44]. It is a communication protocol for industrial automation. One of the strong aspects of this technology is its reliability, which is an obvious advantage considering the typical application requirements in industrial automation and the long life cycle of industrial networks. It allows setting parameters in the network nodes, as well as remotely controlling devices and exchange data. In the HART protocol, as in the four others mentioned above, security is not a part of the framework and thus the protocol can easily be disrupted by an adversary.

4.3.2. Protocols and Standards for Wireless Industrial Networks

Since most existing industrial networks are wired, security has not been considered from the very beginning, as wired networks conventionally are both static and closed. Mobility is seldom required, so the network configuration is usually static and since it is more difficult to join the network illegally, many wired networks are regarded as closed. In contrast, security, especially in terms of encryption, is basically always considered for wireless networks. An overview of open problems in security for industrial wireless networks is presented in [28], where the author mentions efficient key distribution schemes with low overhead and message authentication and integrity checks considering fieldbuses as main open challenges. A wider list of challenges along with the central design approaches for wireless network is presented in [45]. The main directions and emerging challenges for wireless sensor networks applied to the automation domain are highlighted and evaluated in [6]. The coexistence of wired and wireless system was investigated in [46] from a physical layer point of view, and as a case study, the authors consider PROFIBUS, the standard for fieldbus communications in industrial automation.

There are several wireless communication standards used in industrial networks. IEEE 802.11 (WiFi) defines the link and physical layers of the network. One of the main characteristics of the technology is high throughput. However, IEEE 802.11 cannot support real-time requirements since it uses a channel access method based on carrier sensing where random delays may occur, which makes it difficult to use for safety-critical applications. The security solution for the first version of the standard, Wired Equivalent Privacy (WEP) was unsuccessful, and was soon changed to Wi-Fi Protected Access (WPA), and later to WPA2. Each new version solved some issues encountered in the previous one [47]. WPA2 uses the Advanced Encryption Standard (AES) for encryption that can be considered secure with today's standards. It provides such services as authentication, confidentiality and integrity.

Another family of protocols for wireless networks with shorter range is IEEE 802.15.4. This standard also defines the link and physical layers, and was developed for short range communications with low-energy requirements. The protocol standards ZigBee, WirelessHART, and ISA 100.11a are all based on IEEE 802.15.4. ZigBee defines a whole protocol stack, intended for wireless sensor networks. WirelessHART is a wireless complement to the

HART protocol described above. However, in contrast to HART, WirelessHART has built-in security solutions [48]. The classical triad of security objectives CIA is provided here for all communication links. Security is established in three possible modes: end-to-end, peer-to-hop and peer-to-peer. End-to-end security is provided by encryption on the network layer. Peer-to-hop security targets malicious outsiders, and is established by holding a network key and calculating a Message Integrity Code (MIC) for the data link layer payload. Finally, the peer-to-peer mode refers to peer-to-peer sessions and can be established by means of keys management. WirelessHART does however not provide such security objectives as authorization and non-repudiation. ISA 100.11a [49] is designed to support a wide range of wireless industrial plant needs, including process automation, factory automation and RFID. Regarding security, ISA 100.11a has similar solutions as WirelessHART with some differences in keys handling, but in contrast to WirelessHART security solutions in ISA100.11a are not mandatory, and are enabled depending on the network configuration [50].

Chapter 5
Thesis Contributions and Overview of Appended Papers

In this chapter the thesis contributions are highlighted and discussed. This is followed by an overview of all appended research papers that constitute the second part of this thesis. For each paper, a summary, a description of the author's contribution is given, and a record of the publication status is given.

5.1. Thesis Contributions

The first contribution of this thesis is literature review coupled with a detailed problem formulation [Papers A and B], targeting the need to develop a new security framework for heterogeneous industrial networks based on a detailed review of existing security solutions for wired as well as wireless networks. A particular challenge that is addressed is the compliance with real-time requirements.

The second contribution is a way of developing a threat model by putting together the adversary model and goals, as well as the system specification, identifying the most important assets [Paper B]. Specific metrics for comparing and classifying difference security frameworks were introduced. By metrics is meant here the set of requirements to which the framework should provide support, such as support for wired or wireless connections, different types of data traffic or the OSI layer at which the solution operates. Next, the metrics were used to evaluate several existing solutions, providing a well-grounded comparison which enables logical reasoning about different security frameworks and it defines the gaps in the existing solutions that need to be covered in order to secure industrial heterogeneous networks. One of the common characteristics of industrial networks is the joint requirements on reliability and timeliness. Even for non-safety-critical applications, there are still deadlines for messages transactions and if a message misses its deadline, the functionality is lost or loses in quality. This implies that the network should have some kind of schedule that is able to ensure that all messages meet their deadlines. In order to follow the schedule, the network participants should share the same notion of time. This means that they should be synchronized. Clock synchronization is thereby an essential asset for industrial networks. Consequently, it is also an appealing and fruitful target for an adversary, since if it is possible to tamper with clock synchronization, the same technique can be applied to several different networks with different use cases. Therefore, it is more likely that the adversary will invest additional resources into its breaching as the attack technique can be reused.

The third thesis contribution [Paper C] is therefore a detailed investigation of the IEEE 1588 standard for clock synchronization from a security point of view. In order to protect clock synchronization in the system, an investigation on how a possible intruder can breach it, is needed. Two well-known approaches are proposed and combined in this thesis: the possibility to break synchronization with IEEE 1588 by conducting a delay attack, and the possibility to perform a man-in-the-middle attack such that delays can be introduced, using the so-called address resolution protocol (ARP) poisoning technique. This technique explodes the vulnerability in ARP, which is used to define correspondence between IP and MAC addresses The proposed attack is verified using AVISPA [51] and the attack consequences and applicability for industrial networks is investigated.

The fourth contribution [Paper D] is an overview of possible mitigation techniques and countermeasures. All solutions were investigated for applicability in industrial networks. It is proposed to use environmental conditions as a possible way to detect adversary influence in the network. Also, an additional Relaxed mode is introduced as a way of coping with the consequences of slight synchronization breaching. In particular, an investigation of the possibility to use network monitoring and analysis for detecting an attack on clock synchronization provided according to the IEEE 1588 standard is evaluated. The monitor can collect the needed statistics derived from the data traffic patterns and detect an intruder and its influence on the network. Such monitoring is a variant of an Intrusion Detection System (IDS). Based on the detection logic, two types of IDS can be distinguished [52]. In the first case, the IDS has a predefined set of traffic patterns for some known attacks. During its operation, the IDS defines the current network pattern and compares it to the set, and if there is a correlation, it raises an alarm. The clear disadvantage of this approach called *signature based detection* is the inability to detect unknown types of attack. An IDS of the second type has a known set of patterns about the system when in its normal state. In this case, the comparison is between the current network pattern and the predefined set, and an alarm is raised when there is no correlation between the two. This approach is called *anomaly detection* and it can detect unknown attacks, but it will fail if the adversary can learn the system pattern and mask its actions. Several ways of performing delay attacks and its corresponding influence on the network traffic statistics are demonstrated using the AVISPA tool and mathematical analysis. The distributed monitor, collecting and analysing specific statistics about the data traffic, is introduced. The collected data can be used for many purposes, but its applicability from a security point of view is evaluated.

Finally, the fifth and the last contribution of the thesis [Paper E] is a formal analysis of the interactions between an adversary and the network monitor applying a game theoretical approach. The game is formulated by defining its players, together with their strategies and payoffs depending on the outcome. Game theory is a mathematical tool that allows to formally analyze possible interactions between game players and explore their connections to payoff functions and game outcomes. This approach is helpful when investigating a decision-making process of a problem, formulated as a game. Therefore, if correctly applied with valid assumptions allowing to properly formulate the game, game theory can be used to predict the outcome of the game based on the most probable behavior of its players [53]. This approach is widely applied in analyzing security interactions as it is possible to consider parties with contradicting interests. Depending on the application requirements, the game can be zero-sum if the payoffs of players are balanced and sum up to zero, dynamic or static depending on the number of rounds, complete or incomplete depending on the knowledge the players have about each other strategies and actions [54]. In [55] a game theoretic approach was applied to analyze the interactions between an adversary and a sensor network. The authors propose a framework for an IDS based on the derived Nash Equilibriums in the game. The applicability of game theory for IDS was investigated in [56], whereas security issues in vehicular communication were addressed by formulating a corresponding game in [57]. A comprehensive analysis of network security problems and related formulated games is presented in [58]. The authors consider six categories of security techniques and areas, and one of them is IDSs. The possibility to optimize IDSs in the sense of configuration and attack response are examined. The case of applying game theory to wireless ad-hoc networks is presented in [59]. The authors list possible benefits from using the approach, such as a possibility to analyze a distributed system and to get as a result a cross layer optimized solution. Another paper investigating game theory in the scope of mobile ad-hoc networks is [60]. The authors consider nodes that can be normal, selfish or malicious, including an investigation of possible way for a node to cheat, i.e., to not share private information. As the criteria for decision making, not only the Nash Equilibrium was considered, but other options like Pareto optimality. The authors investigate to which degree a node needs to help an opponent, e.g., in a packet forwarding game.

In order to analyse the interactions between the adversary and the network monitor in this work, different probability functions for action strategies are introduced. Furthermore, adaptive rules for switching between strategies in the network are considered. The analysis allows a comparison of adversary strategies from a detection point of view.

5.2. Overview of Appended Papers

Paper A (Chapter 7).
Title: ***Towards Secure Wireless TTEthernet for Industrial Process Automation Applications***
Authors: E. Lisova, E. Uhlemann, J. Åkerberg, and M. Björkman.
Abstract: TTEthernet is a communication platform that builds on Ethernet, but extends it to include fault-tolerance and real-time mechanisms. The existing TTEthernet technology is developed for wired networks. A natural step for improving and extending the current application field is the introduction of a mixed wired and wireless network. However, this step requires research both about possible adaptation of existing systems as well as implementation of new technologies. A central research question is the security aspects of real-time sensor networks using wired and wireless technologies based on TTEthernet. In this paper, we identify and classify the most important aspects to consider in order to provide secure communications in such safety-critical industrial applications and propose a potential solution to address identified issues.
Author's contribution: The author proposed the method for threat modeling and wrote most of the text.

Status: published in the proceedings of the 19th IEEE Conference Emerging Technologies and Factory Automation (ETFA), Barcelona, Spain, Sep., 2014.

Paper B (Chapter 8).
Title: ***A survey of Security Frameworks Suitable for Distributed Control Systems***
Authors: E. Lisova, E. Uhlemann, W. Steiner, J. Åkerberg, and M. Björkman.
Abstract: Nowadays distributed control systems have become more and more common and important in everyday life. However, as many distributed control systems become mobile, wireless, autonomous, ubiquitous and connected, the need for secure communication is imminent. In particular, the need for a general security framework with sufficiently flexible structure, and applicable for various use cases, emerges. Especially this applies to control system based on heterogeneous networks consisting of a wired and a wireless parts. Wired networks are nowadays often connected to Internet and thereby more exposed to potential attackers, and wireless networks are, by nature, more vulnerable to eavesdropping, jamming and hijacking. In this paper we define a scope of use cases based on distributed control, together with requirements for evaluating existing security solutions and frameworks. In addition, several frameworks, mainly from the area of industrial automation, are surveyed and evaluated based on the identified use cases and security requirements.
Author's contribution: The author was the main driver of the paper, wrote the major part of the text and proposed the evaluation technique for the frameworks.
Status: published in the proceedings of the International Conference on Computing and Network communications, Trivandrum, India, Dec., 2015.

Paper C (Chapter 9).
Title: ***Risk Evaluation of an ARP Poisoning Attack on Clock Synchronization for Industrial Applications***
Authors: E. Lisova, E. Uhlemann, W. Steiner, J. Åkerberg, and M. Björkman.
Abstract: Nowadays, mixed wireless and wired networks are used everywhere in everyday life, including in industry where they often support time-critical applications. Industrial applications with high precision requirements are subject to real-time constraints, and thus one of the main assets, regardless of application area, is clock synchronization. Considering such networks, synchronization is the first thing to secure against a possible malicious adversary. In this paper, we consider ARP poisoning as a possible technique to disrupt clock synchronization and evaluate the effects of such an attack on the IEEE 1588 standard. We describe possible ways of performing ARP poison-

ing to disrupt synchronization and survey several mitigation techniques and their applicability within the industrial application area.
Author's contribution: The author was the main driver of the paper and wrote the major part of the text. In addition, the author proposed the possible ways of breaking clock synchronization, i.e. combining ARP poisoning and delay attack, and performed the evaluation using the AVISPA tool.
Status: published in proceedings of the International Conference on Industrial Technology (ICIT), Taipei, Taiwan, Mar., 2016.

Paper D (Chapter 10).
Title: ***Protecting Clock Synchronization, an Adversary Detection by Network Monitoring***
Authors: E. Lisova, M. Gutierrez, W. Steiner, E. Uhlemann, J. Åkerberg, R. Dobrin, and M. Björkman.
Abstract: Today, industrial networks are often used for safety-critical applications with real-time requirements. The architecture of such applications usually has a time-triggered nature that has message scheduling as a core property. Real-time scheduling can be applied only in networks where nodes share the same notion of time, i.e., they are synchronized. Therefore, clock synchronization is one of the fundamental assets of industrial networks with real-time requirements. However, standards for clock synchronization, i.e., IEEE 1588, do not provide the required level of security. This raises the question about clock synchronization protection. In this paper we identify a way to break synchronization based on the IEEE 1588 standard by conducting a man-in-the-middle (MIM) attack followed by a delay attack. MIM attack can be accomplished through e.g., Address Resolution Protocol (ARP) poisoning. Using AVISPA tool we evaluate the potential to perform an ARP poisoning attack. Next, an analysis of the consequences of introducing delays is made, showing both that the attack can, indeed, break clock synchronization and that some design choices, such as a relaxed synchronization condition mode, delay bounding and using knowledge of environmental conditions, can be made to make the network more robust/resilient against these kinds of attacks. Lastly, network monitoring is proposed as a technique to detect anomalies introduced by an adversary performing attacks targeting clock synchronization. The monitoring capabilities are added to the network using a Configuration Agent, which, based on data obtained from the network, is able to detect an attack. The main contribution of the paper is a detailed problem description and evaluation of a security vulnerability in IEEE 1588 against delay attacks together with an evaluation of several approaches as possible mitigation techniques for the attack.

Author's contribution: The author wrote the part of text concerning the system model, the attack description, the possible countermeasures and the attack evaluations as well as one of the mitigation techniques, namely using environmental conditions. In addition, the author proposed the scenarios of the attacks, and did the evaluation of the attacks in AVISPA.
Status: to appear in the Special Issue of the Journal of Electrical and Computer Engineering, in the Special Issue "System and Network Security: Anomaly Detection and Monitoring", Vol. 2016.

Paper E (Chapter 11).
Title: ***Game Theory Applied to Secure Clock Synchronization with IEEE 1588***
Authors: E. Lisova, E. Uhlemann, W. Steiner, J. Åkerberg, M. Björkman.
Abstract: Industrial applications usually have real-time requirements or high precision time-based demands. For such applications, clock synchronization is one of the main assets that needs to be protected against malicious attacks. To provide sufficient accuracy for distributed time-based applications, appropriate techniques for preventing or mitigating delay attacks that breach clock synchronization are needed. In this paper, we apply game theory to investigate possible strategies of an adversary and a network monitor, aiming to detect anomalies introduced by the adversary performing attacks targeting clock synchronization. We investigate the interconnection of payoffs for both sides and propose the quarantine mode as a mitigation technique. Delay attacks with constant, linearly increasing, and randomly introduced delays are considered, and we show that random delays have the highest probability of being detected as they differ the most from delays due to environmental changes.
Author's contribution: The author was the main driver of the paper, wrote most of the text, proposed the idea of appling game theory to model interactions between an adversary and the monitoring system and formulated the game settings.
Status: the shorten version is submitted to 2016 International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS).

Chapter 6
Conclusions

This chapter presents main conclusions of the thesis and points direction for future continuation of the research.

6.1. Summary

As heterogeneous networks, consisting of wireless as well as wired sub-networks, are gaining attention in industry due to increased flexibility and mobility, the need to secure them emerges. The problem of securing industrial networks with real-time requirements and a heterogeneous structure was therefore formalized and a procedure for threat modelling was proposed. The investigation of existing security solutions applied at different layers of the OSI model suggests that a cross-layered structure is appropriate for the developed framework. Given this, a security solution for heterogeneous industrial networks can be developed, such that the adopted security solutions are selected and designed according to the specific requirements from industrial heterogeneous systems. Industrial systems have real-time requirements due to interaction with some physical process and consequently clock synchronization is a fundamental asset that needs to be protected from an adversary. To this end, a vulnerability analysis of the asset synchronization based on the widely deployed IEEE 1588 standard was conducted, and it identified a possibility to break clock synchronization through a combination of a man-in-the-middle attack and a delay attack. This attack is appealing to an adversary as it can target any network requiring synchronization.

Next, a survey of different mitigation techniques and countermeasures was made, which clearly indicates that existing security solutions have problems complying with the requirements for industrial networks in general, such as low communication over-head, and with requirements for safety-critical applications in particular such as absence of single points of failure. It was further shown the problem of detecting if and when clock synchronization is breached, is a challenging task. One of the reasons is the presence of many natural disturbances in the networks, e.g., environmental conditions, causing "natural" delays. However, using prior knowledge about the normal behaviors of nature, such as mean and deviation, several mitigation techniques, such as a relaxed synchronization condition mode, delay bounding and using knowledge of existing environmental conditions, could still be identified, making the network more resilient against these kinds of attacks.

Finally, a network monitor aiming to detect anomalies introduced by the adversary performing attacks targeting clock synchronization was proposed as a mean to detect the delay attack. Distributed monitoring of the inter-arrival times of the messages can detect some specific types of delay attacks and using a set of indicators regarding possible adversary presence in the network, can help the network monitor. Finally, using game theory, a formal analysis of the interaction between an adversary, targeting the clock synchronization via a selective asymmetric delay attack, and the network monitor collecting statistics of measures was made. It was shown that, based on this techniques, it is possible to say which strategy is the most beneficial for the adversary depending on the game configuration. Using this knowledge, the network monitor can deploy appropriate protection techniques.

By securing clock synchronization, a significant step towards developing a complete security framework suitable for all types of industrial heterogeneous networks with real-time requirements has been taken.

6.2. Future Work

There are several directions for future work aiming towards completing the targeted security framework, and these are briefly summarized below.

<u>*Trend detection techniques.*</u>

More comprehensive trend detection techniques can be considered in order to enhance the efficiency and convey a full evaluation of the applicability of the monitor to protect clock synchronization. Applying additional techniques can increase the amount of indicators which potentially can elevate the detection sensitivity and cover more types of attacks. First, a comprehensive literature study should be done to identify the most efficient techniques, and thereafter a joint analysis of new and already considered techniques should be conducted.

Mobility requirement.

Adding mobility to the list of requirements will allow broaden the application area and gain even more benefits from using wireless solutions. The specific challenge is to protect clock synchronization while simultaneously supporting mobility, as moving nodes can change link distances in the network and make asynchronous delay detection even more complicated. Therefore, different techniques targeting this problem should be added to the network monitoring approach.

The next iteration on the way to build security framework.

As was mentioned above, an iterative approach is used when developing the security framework, where each iteration may involve a new asset or a new security objective. Each system assets is first considered separately, and secondly jointly with the previous ones in an overall evaluation of its protection. The idea of using network monitoring for providing the required security services looks promising. Therefore, adding additional functionality to the monitor is the first approach that will be considered. At the current stage of the research, clock synchronization is the scope as a prime asset for industrial heterogeneous networks. The next step implies further investigations of the assets in industrial heterogeneous network and development of appropriate security solutions complying with the identified requirements. The idea is to try to combine future solutions with the ones proposed by now to develop an extended version of the monitor that will have a broader functionality and could cover all required security objectives for the main identified system assets.

Bibliography

[1] D. Moirandi, E. Uhlemann, S. Vitturi, and A. Willig, "Guest Editorial: Special Section on Wireless Technologies in Factory and Industrial Automation, Part I," *IEEE Transactions on Industrial Informatics,* vol. 3, no. 2, pp. 95-98, 2007.

[2] D. Dzung, M. Naedele, T. P.Von Hoff, and M. Crevatin, "Security for Industrial Communication Systems," *Proc. IEEE,* vol. 93, no. 6, pp. 1152-1177, 2005.

[3] S. Raza and T. Voigt, "Interconnecting WirelessHART and legacy HART networks," presented at the Distributed Computing in Sensor Systems Workshops (DCOSSW), 2010 6th IEEE International Conference on, Santa Barbara, CA, 2010.

[4] IEEE. (2008). *IEEE 1588, "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems"*. Available: http://www.nist.gov/el/isd/ieee/ieee1588.cfm, 8 April 2016.

[5] H. J. Holz, A. Applin, B. Haberman, D. Joyce, H. Purchase, and C. Reed, "Research methods in computing: what are they, and how should we teach them?," Proceeding ITiCSE-WGR '06 Working group reports on ITiCSE on Innovation and technology in computer science education 2006.

[6] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," *in Proc. INDIN*, Portugal, July, 2011, pp. 410-415.

[7] V. Mhatre and C. Rosenberg, "Homogeneous vs heterogeneous clustered sensor networks: a comparative study," *in Proc. IEEE International Conference on Communications*, Paris, France, 20-24 June, 2004, pp. 3646 - 3651.

[8] M. Valero, S. S. Jung, A. S. Uluagac, Y. Li, and R. Beyah, "Di-Sec: A distributed security framework for heterogeneous Wireless Sensor Networks," *in Proc. annual International Conference on Computer*

communications (INFOCOM), Orlando, Florida USA, 25 - 30 March, 2012, pp. 585-593.

[9] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer, "The Time-Triggered Ethernet (TTE) design " *in Proc. International Symposium on Object-Oriented Real-Time Distributed Computing*, Seattle, USA, May, 2005, pp. 22-33.

[10] M. Cheminod, L. Durante, and A. Valenzano, "Review of Security Issues in Industrial Networks," *IEEE TII,* vol. 9, no. 1, pp. 277-293, Feb 2013.

[11] E. Lisova, E. Uhlemann, W. Steiner, J. Åkerberg, and M. Björkman, "A Survey of Security Frameworks Suitable for Distributed Control Systems," *in Proc. International conference on computing and Network Communications (CoCoNet)*, Technopark, Trivandrum, India, 16-19 Dec., 2015, pp. 205-211.

[12] M. Anand, E. Cronin, M. Sherr, M. Blaze, Z. Ives, and I. Lee, "Sensor network security: more interesting than you think," *in Proc. USENIX HotSec*, San Jose, CA, Jul, 2006, pp. 5-5.

[13] M. Ullmann and M. Vögeler, "Delay attacks - implication on NTP and PTP time synchronization," *in Proc. International Symposium on Clock Synchronization for Measurenment, Control and Communication (ISPCS)*, Brescia, Italy, Oct., 2009.

[14] H. Kopetz and W. Ochsenreiter, "Clock Synchronization in Distributed Real-Time Systems," *IEEE Transactions on Computers,* vol. C-36, no. 8, pp. 933-940, 1987.

[15] E. Lisova, E. Uhlemann, W. Steiner, J. Åkerberg, and M. Björkman, "Risk Evaluation of an ARP Poisoning Attack on Clock Synchronization for Industrial Applications," *in Proc. IEEE International Conference on Industrial Technology (ICIT)*, Taipei, Taiwan, 14-17 Mar., 2016.

[16] V. Stavridou and B. Dutertre, "From Security to Safety and Back," *in Proc. Computer Security, Dependability and Assurance: From Needs to Solutions*, York, UK1998, pp. 182-195.

[17] A. Avižienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing,* vol. 1, no. 1, pp. 11-33, 2004.

[18] N. Kuntze, C. Rudolph, B. Brisbois, and M. Boggess, "Security vs. safety: Why do people die despite good safety?," *in Proc. Integrated Communication, Navigation, and Surveillance Conference (ICNS)*, Herdon, VA, 21-23 Apr., 2015, pp. A4-1 - A4-10.

[19] A. Chakrabarti and G. Manimaran, "Internet infrastructure security: a taxonomy," *IEEE Networks,* vol. 16, no. 6, pp. 13-21, Nov. 2002.

[20] D. Papp, Z. Ma, and L. Buttyan, "Embedded systems security: Threats, vulnerabilities, and attack taxonomy," presented at the 13th Annual Confernece onPrivacy, Security and Trust (PST), Izmir, 2015.

[21] D. Welch and S. Lathrop, "Wireless security threat taxonomy," presented at the IEEE Systems, Man and Cybernetics Society Information Assurance Workshop, 2003.

[22] P. Savolainen, E. Niemela, and R. Savola, "A Taxonomy of Information Security for Service-Centric Systems," *in Proc. EUROMICRO Conference on Software Engineering and Advanced Applications*, 28-31 Aug., 2007, pp. 5-12.

[23] S. Zafar and R.G. Dromey, "Integrating safety and security requirements into design of an embedded system," *in Proc. 12th Asia-Pacific Software Engineering Conference (APSEC)*, 15-17 Dec., 2005.

[24] R. Oates, D. Foulkes, G. Herries, and D. Banham, "Practical Extensions of Safety Critical Engineering Processes for Securing Industrial Control Systems," *in Proc. 8th IET International System Safety Conference incorporating the Cyber Security Conference*, Cardiff, 16-17 Oct., 2013, pp. 1-6.

[25] K. Psounis, "Active networks: Applications, security, safety, and architectures," *IEEE Communications Surveys,* vol. 2, no. 1, pp. 2-16, 1999.

[26] J. Leiwo and S. Heikkuri, "An analysis of ethics as foundation of information security in distributed systems," *in Proc. Thirty-First Hawaii International Conference on System Sciences*, 6-9 Jan., 1998, pp. 213-222.

[27] A. Thekkilakattil and G. Dodig-Crnkovic, "Ethics Aspects of Embedded and Cyber-Physical Systems," *in Proc. IEEE 39th Annual International Computers, Software & Applications Conference*, Taichung, 1-5 Jul., 2015, pp. 39-44.

[28] A. Willig, "Recent and Emerging Topics in Wireless Industrial Communications: A Selection," *IEEE TII,* vol. 4, no. 2, May 2008.

[29] Z. Drias, A. Serhrouchni, and O. Vogel, "Analysis of cyber security for industrial control systems," *in Proc. International Conference on Cyber Security of Smart Cities (SSIC)*, Shanghai, 5-7 Aug., 2015, pp. 1-8.

[30] V. Delgado-Gomes, J. F. Martins, C. Lima, and P. N. Borza, "Smart grid security," *in Proc. International Conference on Compatibility*

and Power Electronics (CPE), Costa da Caparica, 24-26 Jun., 2015, pp. 534-538.

[31] V. Kounev, D. Tipper, A. A. Yavuz, B. M Grainger, and G. F Reed, "A Secure Communication Architecture for Distributed Microgrid Control," *IEEE Transactions on Smart Grid,* vol. 6, no. 5, pp. 2484-2492, 2015.

[32] M. Bere and H. Muyingi, "Initial Investigation of Industrial Control System (ICS) Security Using Artificial Immune System (AIS)," *in Proc. International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, Windhoek, 17-20 May, 2015, pp. 79-84.

[33] R. M. John, J. P Cherian, and J. Kizhakkethottam, "A Survey of Techniques to Prevent Sybil Attacks," *in Proc. International Conference on Computer Communication and Informatics*, Coimbatore, 8-10 Jan., 2015.

[34] R. Bhumkar and D.J. Pete, "Reduction of Error Rate in Sybil Attack Detection for MANET," *in Proc. International Conference on Intelligent Systems and Control*, Coimbatore, 9-10 Jan., 2015.

[35] K. Koscher, A. Czeskis, F. Roener, and S. Patel, "Experimental Security Analysis of a Modern Automobile," *in Proc. IEEE Sym. SP*, Oakland, CA, May, 2010, pp. 447 - 462.

[36] K. Allenby and T. Kelly, "Deriving safety requirements using scenarios," presented at the Fifth IEEE International Symposium on Requirements Engineering, Toronto, Ont., 2001.

[37] P. Neumann, "Communication in industrial automation—What is going on?," *Control Engineering Practice,* vol. 15, no. 11, pp. 1332-1347, 2007.

[38] S. M Rinaldi, J. P Peerenboom, and T. K Kelly, "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies," *IEEE Control Systems Magazine,* vol. 21, no. 6, pp. 11-25, 2001.

[39] B. Galloway and G. P. Hancke, "Introduction to Industrial Control Networks," *IEEE Communications Surveys & Tutorials,* vol. 15, no. 2, pp. 1-21, 2012.

[40] P. Gaj, J. Jasperneite, and M. Felser, "Computer Communication Within Industrial Distributed Environment—a Survey," *IEEE TII,* vol. 9, no. 1, pp. 182-189, Jul 2012.

[41] *Modbus Protocol Specification*. Available: http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf, 8 April 2016.

[42] *Distributed Network Protocol*. Available: http://www.dnp.org, 8 April 2016 .

[43] W. Steiner, G. Bauer, B. Hall, and M. Paulitsch, "*TTEthernet: Time-Triggered Ethernet*," in *Time-Triggered Communication*, R. Obermaisser, Ed., ed CRC Press, IDate, 2011.

[44] *HART protocol*. Available: http://en.hartcomm.org/hcp/tech/aboutprotocol/aboutprotocol_specs.html, 8 April 2016.

[45] V.C. Gundor and G.P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *IEEE TIE,* vol. 56, no. 10, pp. 4258-4265, 2009.

[46] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Networks," *Proc. IEEE,* vol. 93, no. 6, pp. 1130-1151, June 2005.

[47] A. H. Lashkari and M. M. S. Danesh, "A survey on Wireless Security protocols (WEP WPA and WPA2/802.11i)," *in Proc. Computer Science and Information Technology ICCSIT*, Beijing, China, 8-11 Aug. 2009, 2009, pp. 848-52.

[48] S. Raza, A. Slabbert, T. Voigt, and K. Landernas, "Security considerations for the WirelessHart protocol," *in Proc. ETFA*, Mallorca, Spain, Sep, 2009, pp. 1-8.

[49] *ISA100, Wireless Systems for Automation*. Available: http://www.isa.org/isa100, 8 April 2016.

[50] S. Peterson and S. Carlsen, "WirelessHART versus ISA100. 11a: the format war hits the factory floor," *IEEE Industrial Electronics Magazine,* vol. 5, no. 4, pp. 23-34, 2011.

[51] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. Von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Vigano, and L. Vigneron, "The AVISPA tool for the automated validation of internet security protocols and applications," *in Proc. 17th International Conference on Computer Aided Verification (CAV)*, Edinburgh, Scotland, Jul., 2005.

[52] M. Garuba and D. Fraites, "Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems," *in Proc. Fifth International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, NV2008, pp. 592-598.

[53] M. Wooldridge, "Does Game Theory Work?," *IEEE Intelligent Systems,* vol. 27, no. 6, pp. 76-80, Nov.-Dec. 2012.

[54] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A Survey of Game Theory as Applied to Network Security," *in*

Proc. 43rd Hawaii international Conference on System Sciences (HICSS), Honolulu, HI 2010, pp. 1-10.

[55] A. Agah, Sa. K. Das, K. Basu, and M. Asadi, "Intrusion detection in sensor networks: a non-cooperative game approach," *in Proc. Third IEEE International Symposium on Network Computing and Applications (NCA)*2004, pp. 343-346.

[56] T. Alpcan and T. Basar, "A game theoretic approach to decision and analysis in network intrusion detection," *in Proc. 42nd IEEE Conference on Desicion ans Control* Hawaii, USA, 9-12 Dec., 2003, pp. 2595-2600.

[57] S. Buchegger and T. Alpcan, "Security Games for Vehicular Networks," *Mobile Compiting, IEEE Transaction on,* vol. 10, no. 2, pp. 280-290, 2011.

[58] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR),* vol. 45, no. 3, 2013.

[59] V. Srivastava, J. Neel, A. B. Mackenzie, P. Menon, L. A. Dasilve, J. E. Hicks, J. H. Reed, and R. P. Gilles, "Using Game Theory to Analyze Wireless Ad Hoc Networks," *IEEE Communications Surveys & Tutorials,* vol. 7, no. 4, pp. 46-56, 2005.

[60] W. Yu and K. J. R. Lui, "Game Theoretic Analysis of Cooperation Stimulation and Security in Autonomous Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing,* vol. 6, no. 5, pp. 507-521, 2007.

II
Included Papers

Chapter 7

Paper A: Towards Secure Wireless TTEthernet for Industrial Process Automation Applications

Elena Lisova, Elisabeth Uhlemann, Johan Åkerberg, and Mats Björkman.

In Proceedings of 19th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Barcelona, Spain, September 2014.

Abstract

TTEthernet is a communication platform which builds on Ethernet, but extends it to include fault-tolerance and real-time mechanisms. The existing TTEthernet technology is developed for wired networks. A natural step for improving and extending the current application field is the introduction of a mixed wired and wireless network. However, this step requires research both about possible adaptation of existing systems as well as implementation of new technologies. A central research question is the security aspects of real-time sensor networks using wired and wireless technologies based on TTEthernet. In this paper, we identify and classify the most important aspects to consider in order to provide secure communications in such safety-critical industrial applications and propose a potential solution to address identified issues.

7.1. Introduction

Time-Triggered Ethernet (TTEthernet) is [1] a technology that allows extending Ethernet so that it can conform to applications where time-scheduling and predictability are the prime issues. To achieve this extension, several message traffic classes are used: Rate-Constrained (RC), Best-Effort (BE) and Time-Triggered (TT) traffic. TTEthernet allows partitioning all data into these different categories, all with different traffic policies and different temporal characteristics, such that safe and predictable communication for mixed-criticality systems can be established [1]. For instance, TT messages have the highest priority and therefore they are dispatched according to a predetermined schedule, and thus this traffic class is suitable for time-critical data. One significant TTEthernet feature is that it can be used for applications with different time and safety requirements. This way critical applications can co-exist with best effort services without significant interference. Initially, TTEthernet was developed as wired system, but as more and more diverse application requirements emerge, there is a strong market need to make it mixed wireless and wired.

Wireless sensor networks have a number of evident advantages such as mobility, weight, size, simplicity and a list of others depending on the specific application. Hence, its implementation can increase the applicability area, especially in such domains as aerospace, automotive and industrial automation [2]. However, introducing wireless access also gives rise to new problems. Wireless links can more easily be intercepted and influenced as they open up communication also with intruders and eavesdroppers. Consequently, when considering a wireless version of TTEthernet for use in safety-critical applications with real-time requirements, security becomes a prime issue [2, 3]. A revised threat model is needed, the assets must be identified, and application specific security objectives should be defined.

When attempting to solve the problem of developing a secure wireless version of TTEthernet, existing standards and protocols for wired TTEthernet should first be considered [4]. Further, there are a number of wireless standards such as WiFi, WirelessHART and ZigBee, all of them with different security concepts and techniques. An analysis and comparison of these protocol standards can help developing a contemporary security framework that will reflect all necessary wireless features. In particular, we need to consider which threats that are specific for wireless systems; whether the assets set remains the same; which vulnerabilities that should be taken into account; and finally which changes should be introduced into the existing architectural concept for TTEthernet networks.

The main contribution in this paper is a detailed problem formulation targeting the development of a new security framework for mixed wireless and wired networks based on TTEthernet when used in industrial applications. In addition, we suggest a potential approach to solve the identified problems. The challenge that is especially addressed is achieving a standardized security framework that conforms to real-time demands, i.e. that the security mechanisms do not impact the real-time communication functionality, or else the remedy would itself become a denial-of-service attack [5].

The remainder of the paper is organized as follows. In Section 7.2 we consider the threat model by specifying the application requirements, the assets, the adversary goals and the adversary model. Section 7.3 presents an approach for defining essential system features that should be considered within possible security framework development. Section 7.4 describes one possible solution namely Internet Protocol Security (IPSec), together with its drawbacks and ways of overcoming them. Finally, in Section 7.5 we represent our conclusions.

7.2. Threat Model

The threat model should reflect real adversary possibilities which are usually connected to the specific application, since this directly specifies the adversary goals, i.e., what an intruder targets, together with the assets that needs to be protected. Therefore the threat model should also include an adversary model. By adversary model we consider the specific features of an intruder, which can help understanding the possibilities of the intruder and structure ways to characterize the influence of an intrusion.

A. Application Specification

We mainly target industrial applications due to their requirements on dependable transmission of real-time data. Nowadays sensor networks are widely used in industry for controlling, measuring and aggregation of data. For instance, a typical application example from process automation is a paper machine. To produce paper of required quality the humidity of the process should be measured continuously, as deviations from the specified values can lead to quality degradation. As a result real-time requirements are imposed by the system, and the humidity characteristic should be transmitted timely, reliable and continuously. Due to the speed of the paper production process, it is safer to measure humidity with wireless sensors rather than using a wired solution as wireless sensor can be fitted directly into a fast rotating part of the machine, rather than manually measuring using wired sensors.

B. Adversary Goals

The next step is setting the adversary goals and based on these, try to analyze what consequences it can have if the goals are reached and what countermeasures that can be imposed. According to [6], the three main adversary goals for sensor networks are disruption, eavesdropping and hijacking. As follows from the application example mentioned above, eavesdropping is not terminal for the considered sensor network. If an intruder can get information from the sensors (e.g. a water percentage in a specific type of paper) it can be objectionable as it potentially is a production secret but not fatal. On the other hand, if the intruder can change the data from the sensors or damage the system it may have terminal consequences. Therefore, eavesdropping can be eliminated from the list of adversary goals mentioned above. Consequently, the most significant adversary goals are disruption and hijacking as their impact on the targeted application is considerable.

C. Assets

When considering assets in applications with real-time constraints, one of the main assets is clock synchronization. Mazrahi provides a good classification of possible treats for clock synchronization in [7]: interception and modification; spoofing; replay; rogue master; interception and removal; delay manipulation; denial of service (DoS) attacks for OSI model layers 2 and 3; cryptographic performance; time source spoofing. This classification is very general and can be used regardless of if we consider a wired, wireless or mixed system. The specific set of tools needed to protect the asset clock synchronization depends on the system structure.

D. Adversary Model

In general, all adversary aspects mentioned in [8] are of importance for wireless TTEthernet applications. In particular, the most important adversary features in this context are: whether or not it is passive or active, static or adaptive, an insider or an outsider, the adversary mobility, communication capabilities and computational power.

A passive adversary is a prerequisite for an active one as at first, the adversary passively collects data and then, based on its analysis, starts to actively influence. If the adversary is an outsider, the goal is more connected to system disruption, as it is easier to suppress the channel or cause interference rather than to hijack the whole system. If the adversary is an adaptive one and can change its behavior depending on network response, it is more dangerous for the system functionality. Communication capabilities reflect whether the adversary acts through the network protocols or through the wireless channel or both. Possible adversary computational power depends on the specific application and the value of its assets. The higher the potential gain of interfering or hijacking a system, the more likely it is that the adversary invests in more computational power.

7.3. Approach

After defining the threat model we can propose an approach that includes a list of security objectives. By security objectives, we consider system features that should be imposed and according to which we can choose a protocol that can cover all, or almost all, requirements needed. Therefore we need to analyze the assets, i.e., the things we want to protect. We should identify in which way an adversary can gain possession of or control over the assets such that we can identify the objective of the security mechanisms that are introduced. The set of security objectives that should be introduced depends on the application (e.g. military or personal data). Initially, we consider all objectives mentioned in [8] and [9] and thereafter, we remove the ones that are not directly applicable for process automation applications.

Confidentiality. Generally this security objective refers to that the adversary must not know information from the sensors. However, according the identified adversary goals, confidentiality is not strictly required for considered application, as the level of paper humidity data is no secret, but rather a well-known fact used as a feedback to control the process.

Integrity. This notion is connected with the following questions: has the data been corrupted; can we trust this source? Integrity is a prime issue for the asset clock synchronization in applications with real-time constrains.

Authentication. By this we consider that we must know from whom (which network node) we get this information. Consequently, authentication is also a key point for the clock synchronization asset.

Availability. Mainly this objective refers to the fact that the service provided must be available, i.e., in our case, the paper making process must function all hours of the day since paper machines are too expensive to stand still. When considering our safety-critical application, availability is possibly the most important objective.

Anonymity. This notion usually is understood as the possibility to use a network without being identified or having private data shared without consent. As we consider sensor networks in process automation, this is not the most important security objective.

Auditability. System behavior reconstruction which is assured by auditability can help to enhance reliability if it is made adaptive, but if we are talking about safety-critical applications, it is not a prime issue as such systems should be as reliable as possible already from the beginning. Note that a sensor network used for controlling a paper machine can be considered safety-critical from the money loss point of view.

Nonrepudiability. This objectivity is about liability and has more legal than safety consequences, and therefore it is not a prime issue for the targeted application.

Third-party protection. This is about preventing damage done to third parties and it also more connected with reputation and legal consequences and therefore out of scope here.

Conformance. The network should work in accordance to the protocol. Just as with availability, this is one of the most important objectives for safety-critical applications.

After analyzing the security objectives mentioned above we can conclude that the ones that are most important for our considered applications are conformance, availability, integrity and authentication. Based on the list of identified objectives, we determine suitable techniques and approaches that can be used as a basis for the security framework. It is reasonable to first analyze existing protocols and techniques, to establish whether or not they can be of use for the identified set of targets above.

Two additional points also should be considered. Firstly, since TTEthernet is entirely compatible with Ethernet, we should evaluate protocols that can provide secure communications based on the Internet Protocol (IP). Secondly, we should look at the security analysis of wired TTEthernet [4], made by Steiner. The article provides some drawbacks and points to pay attention to in existing wired TTEthernet solution and also offers possible candidates for future developing. As shown in [4], systems based on TTEthernet can be vulnerable to internal attacks (e.g. replay attack). As possible security extensions replay protection, messages authentication and frame encryption are suggested.

7.4. Potential Solution: IPSec

A possible candidate for the security framework of wireless TTEthernet is IPSec. IPSec [10] is a set of protocols which can provide the following types of protection: user data encryption, replay attack protection, and message integrity authentication. Devices in an IPSec network jointly decide which technique is needed according to their individual requirement specification.

The two main protocols that are used in IPSec are Authentication Header (AH) and Encapsulating Security Payload (ESP). The first one provides message integrity, data origin authentication and protection against replay attacks. The second one allows encrypting the whole datagram, using a set of encryption and authentication algorithms. Also one of the key points for IPSec is Internet Key Exchange (IKE). Mainly this is a combination of three protocol functions Internet Security Association (SA) and Key Management Protocol (ISAKMP) – a key exchange method; Secure Key Exchange Mechanism for Internet (SKEME) which provides public key encryption; and OAKLEY which contains specific key exchanging mechanisms for different key exchange modes. It is important to understand how IKE works, as the security objectives mentioned above only can be achieved only when this protocol is applied correctly.

All in all, IPSec provides a wide range of mechanisms for security which all depend on the specific combination of modes, IP versions and protocols. It complies with several of the above listed security objectivities. However, it also has several evident drawbacks. Firstly, IPSec is initially oriented towards point-to-point connections, so if used it in broadcast scenarios, adjustments are needed. Secondly, IPSec was not initially developed for application with real-time requirements. As these two problems appear as the two main showstoppers for relying only on IPSec as security protocol for mixed wireless and wired TTEthernet, the paper considers some suggestions on how to overcome them.

A. Solutions for Multicast

There exist some solutions for adapting IPSec to multicast mode, e.g., the Generic Route Encapsulation (GRE) protocol developed by Cisco. GRE is a tunneling protocol, which allows encapsulating protocols inside virtual point-to-point links over IP based internetworks using IPSec. Encapsulation in GRE can be performed on an arbitrary level, it is very general and allows a system that needs to transfer a packet to first encapsulate it in a GRE packet and next, this packet can again be encapsulated in yet another protocol and transferred through the GRE tunnel.

Another possible solution, also developed by Cisco, is called Dynamic Multipoint IPSec VPN (DMVPN), which can be described as a multipoint GRE (mGRE) together with the Next Hop Resolution Protocol (NHRP). NHRP is a protocol that dynamically can maps non-broadcast multi-access networks. Basically NHRP allows two functions, the first one is the possibility to allow a Next Hop Client dynamically registered with a Next Hop Server and the second one is the possibility for one client to dynamically find the mapping between the logical VPN IP and the physical IP of another client within the same network. Therefore when DMVPN is applied, IPSec is used as an encryption function, GRE or mGRE is used for setting up a tunnel and finally NHRP dynamically addresses different problems that may arise.

The techniques mentioned above can all be used for solving the IPSec multicast problem.

B. Solutions for Real-Time

When applying IPSec to TTEthernet, there are two main features that lead to problems with real-time requirements [11]. The first one is the increased packet size required by the additional IPSec headers and the second one is the time or complexity that must be spent on data encryption and decryption. There are several research results that show how the implementation of IPSec can affect the quality of service (QoS) characteristics of the systems [12], [13]. However, there are also several approaches available that can improve its QoS performance. IPSec performance can, for example, be improved by introducing the Multi-Layer IPSec (MLIPSec) protocol [14], which allows intermediate devices to decrypt parts of datagrams in order to speed up the routing process. However the possibility to use this method depends on the network size as well as its configuration and thus in some cases this approach is inapplicable. Furthermore MLIPSec is only suitable for static environments. Another approach, developed by Choi [15], is Mobile Multi-Layered IPSec (MML-IPSec). This approach includes an efficient key distribution protocol and also two mobile protocols. This technique is developed specifically for wireless communication and offers a dynamic version of MML-IPSec that allows varying security levels depending on data significance. The key distribution protocol includes mobility support, in the form of two protocols: Proactive Key Distribution (PKD) protocol and Dynamic Key Migration (DKM) protocol. The first one pre-establishes the SA with the current foreign agent and its neighbors, whereas the second one helps the SA to migrate between foreign agents while its user is moving.

It is important to note that increased packet size and increased delay due to encryption are features that mainly affect the delay in a predictable manner, which can be taken into account by the real-time scheduler. However, a fully loaded schedule may include task sets that are no longer schedulable if each packet requires more time to transmit and process. Alternatively, the overhead implied by IPSec may require longer time slots in a time-triggered setting which increases the overall superframe length. To determine if IPSec completely fulfills the real-time requirements of an existing sensor network, the network load should be estimated, which again will make its usefulness highly application specific. Also it is obvious that it is not enough to find a solution to each isolated problem, but also the combination of different protocols must be considered.

7.5. Conclusions and Future Work

In this paper, we have defined important aspects to consider when developing a security framework targeting applications based on wireless TTEthernet. As many of the solutions are rather application specific, we used process automation as a possible use case due to its inherent real-time requirements, the market drive to introduce wireless access technologies in industry and the need to be compliant with existing wired time-triggered networks. Based on this field of applications, we identified adversary goals, adversary models and system assets. In addition, we outlined an approach on how to address the identified issues, which includes a list of security objectives. Finally, IPSec was investigated as a possible solution, its drawbacks were listed and also different ways to overcome these drawbacks were proposed.

In future work we will further investigate how IPSec can be used for the security framework of wireless TTEthernet and how we can adapt and develop this protocol in accordance with the proposed techniques for real-time and multicast requirements.

Acknowledgments

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n°607727. E.

Bibliography

[1] W. Steiner, G. Bauer, B. Hall, and M. Paulitsch, "TTEthernet: Time-Triggered Ethernet," in *Time-Triggered Communication*, R. Obermaisser, Ed., CRC Press, August 2011.

[2] Gundor, V.C. and G.P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *Industrial Electronics,* vol. 56, 10, p. 4258-4265, 2009.

[3] S. Raza, A. Slabbert, T. Voigt, and K. Landernas "Security considerations for the WirelessHart protocol," *in Proc. Emerging Technologies and Factory Automation*, Mallorca, Spain, September 2009, pp. 1-8.

[4] W. Steiner, "Candidate security solutions for TTEtnernet," *in Proc. Digital Avionics Systems Conference*, East Syracuse, NY, USA, October 2013, pp. 1-10.

[5] J. Akerberg, M. Gidlund, and M. Bjorkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," *in Proc. 9th IEEE International Conference on Industrial Informatics (INDIN)*, Caparica, Lisbon, July 2011, pp. 410-415.

[6] M. Anand, E. Cronin, M. Sherr, M. Blaze, Z. Ives, and I. Lee, "Sensor network security: more interesting than you think" *in Proc. USENIX Workshop on Hot Topics in Security*, CA, USA, July 2006, pp. 5-5.

[7] T. Mizrahi, "Time Synchronization Security using IPsec and MACsec," *in Proc. 2011 International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Munich, Germany, September 2011, pp. 38-43.

[8] J.A. Clark, J. Murdoch, J.A. McDermid, S. Sen, H.R. Chivers, O. Worthington, and P. Rohatgi, "Threat modelling for mobile Ad Hoc and sensor networks," *in Proc. Second International Confrence on Internet Technologies and Applications*, Wrexham, Nort Wales, UK, September 2007, pp.25-27.

[9] D. Dzung, M. Naedele, T.P.V. Hoff, and M. Crevatin, "Security for Industrial Communication Systems," *PIEEE - Proceeding of the IEEE*, vol. 93, 6, pp. 1152-1177, 2005.

[10] S. Kent and S. Seo, *Security Architecture for Internet Protocol,* IETF RFC 4301, December 2005, http://www.rfc-editor.org/rfc/pdfrfc/rfc4301.txt.pdf

[11] R. Barbieri, D. Brusci, and E. Rosti, "Voice over IPsec: analysis and solutions," *in Proc. Computer Security Applications Conference*, Las Vegas, NV, USA, December 2002, pp. 261-270.

[12] B. Vaidya, J.W. Kim, J.Y. Pyun, J.A. Park, and S. Han, "Perfomance Analysis of Audio Streaming in Secure Wireless Access Network," *in Proc. International Conference on Computer and Information Science*, Jeju Island, South Korea, July 2005, pp. 556-561.

[13] O. Adeyinka, "Analysis of IPsec VPNs performance in a multimedia environment," *in Proc. International Conference on Intelligent Environments*, Seattle, WA, USA, July 2008, pp.1-5.

[14] Y. Zhang and B. Singh, "A multi-layer IPsec protocol" *in Proc. 9th USENIX Secutiry Symposium*, Denver, Colorado, August 2000, pp. 1-17.

[15] H. Choi, H. Song, G. Cao, and T.F.L. Porta, "Mobile Multi-layered IPsec," *Wireless Networks*, vol. 14, 6, pp. 895-913, 2008.

Chapter 8

Paper B: A survey of Security Frameworks Suitable for Distributed Control Systems

Elena Lisova, Elisabeth Uhlemann, Wilfried Steiner, Johan Åkerberg, and Mats Björkman.

In Proceedings of International Conference on Computing and Network Communications (CoCoNet), Trivandrum, India, December 2015.

Abstract

Nowadays distributed control systems have become more and more common and important in everyday life. However, as many distributed control systems become mobile, wireless, autonomous, ubiquitous and connected, the need for secure communication is imminent. In particular, the need for a general security framework with sufficiently flexible structure, and applicable for various use cases, emerges. Especially this applies to control system based on heterogeneous networks consisting of a wired and a wireless parts. Wired networks are nowadays often connected to Internet and thereby more exposed to potential attackers, and wireless networks are, by nature, more vulnerable to eavesdropping, jamming and hijacking. In this paper we define a scope of use cases based on distributed control, together with requirements for evaluating existing security solutions and frameworks. In addition, several frameworks, mainly from the area of industrial automation, are surveyed and evaluated based on the identified use cases and security requirements.

8.1. Introduction

Applications using distributed control are becoming more and more frequent. Examples can be found in areas as diverse as aerospace, automotive, automated factories, chemical processes, civil infrastructure, energy, healthcare, robotic networks, manufacturing, transportation, entertainment, and consumer appliances. The most common form is a set of embedded systems, communicating through some type of network. Rather than having a centrally located device controlling all the embedded subsystems that are part of the control system, each embedded subsystem controls its own operation in a distributed fashion. Nowadays, the possibility to merge several different heterogeneous subsystems into one distributed control system (DCS) is an important requirement [1]. Heterogeneity implies coexistence of many different types of nodes, traffic classes and/or communication links. Networks dealing with different types of nodes and traffic classes can support applications with different criticality levels and be more flexible and efficient. To meet current market demands, more and more technologies are also targeting a wireless or mixed wired and wireless solution [2]. Wireless networks have some evident advantages for DCSs, such as mobility and simplicity for the industrial automation area, and weight and size for the automotive area. Wireless solutions can potentially widen and enhance the application areas [3], but come at a price as wireless channels easier can be influenced and affected by malicious intruders.

As security risks are becoming a showstopper for deployment of DCSs, it is considered as an increasingly important requirement [4]. Security risks exist if there is a vulnerability and a threat. A vulnerability is simply the opportunity to cause damage, and it can be due to a design flaw, an implementation flaw or some weaknesses in terms of oversimplified passwords or keys [5]. A threat exists if there is some value in breaking the system. The term “security” covers a wide range of provided services, so-called security objectives, ranging from a parity code to detect compromised data, via encryption to compromised node detection. A security objective describes what type of threat the system needs to be secured against. Different DCSs have different security objectives. In some networks, data is not confidential and all that is needed is data origin checking, whereas in other networks it is crucially important to keep data confidential, due to e.g., the need to protect product recipes, and in this case encryption is needed. However, since all DCSs are time-critical, they all imply some kind of scheduling [6], and thus it is possible to cause system disruption within a DCS by targeting the clock synchronization functionality. There are many ways of influencing the synchronization, but one of the most difficult to detect and counteract is the delaying attack, as the adversary does not need to alter any data, but only to delay a few synchronization messages to put a node into unsynchronized mode [7].

The main contribution of this paper is a detailed investigation of existing security solutions and frameworks to determine their applicability to heterogeneous DCS. Such an investigation simplifies the choice of technology already at the design stage of new DCS and allows identifying security gaps in existing solutions. To this end, we also propose an approach for comparing and evaluating the suitability of existing security frameworks for the specific security requirements derived from DCSs. This allows a well-founded comparison.

The possible range of use cases for DCS is extremely wide, and they all have different application areas, purposes and ways of realization. However, we target a general profile of use cases based on DCS with similar core characteristics specifically related to distributed control, namely systems requiring high reliability, timeliness and availability, and where something can be controlled from a distance if security is breached. The general profile therefore includes heterogeneous networks employed in different environments, having different topologies and including links of different qualities. Also it covers various types of use cases, e.g., factories and plants with limited physical access possibilities for intruders or private cars with unpredictable human behavior. However, we target security use cases and requirements suitable for generic DCS, where the main characteristics are real-time constraints, reliability, availability, heterogeneity and ability to support different traffic classes. By security framework we consider a proposal in which several different security objectives are achieved. As security is a multifarious term, there cannot be one single separate solution covering all emerging DCS demands. Therefore, an aggregation of several solutions should be considered to achieve a framework with appropriate quality of security system performance.

The remainder of the paper is organized as follows. Section 8.2 describes important terminology for communication security, whereas Section 8.3 presents the scope of use cases within distributed control. In Section 8.4, the system model is described and in Section 8.5 requirements for the considered frameworks are investigated. Sections 8.6 and 8.7 present our evaluation of existing security solutions and frameworks respectively, and, finally, Section 8.8 concludes the paper.

8.2. Components of Communication Security

To evaluate different security frameworks, we need to define the main terminology. In particular, we consider the following components, organized as proposed in Fig. 8.1. System assets are the features that we want to protect in the system. They are the main assets of the use case, as they affect its workflow most and have the highest value. For example, in a system with sensitive data, the data confidentiality is an asset. Adversary goals represent the possible targets of a potential malicious intruder. If we consider intrusion detection systems in plants, a possible adversary goal is system hijacking. An *adversary model* represents a set of possible adversary features, such as geographic location, time and budget. The adversary model is extremely important for correct risk estimation and appropriate security design.

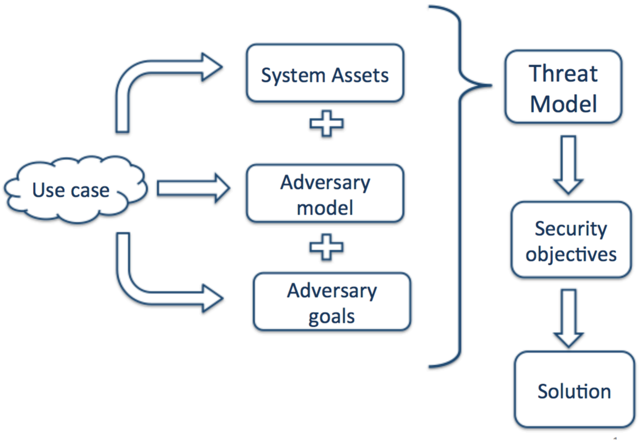

Fig. 8.1 An approach for security framework derivation.

The system assets, the adversary model and goals can all be derived from the considered use case, as shown in Fig. 8.1 An approach for security framework derivation., since they reflect specific properties of the use case. Together, these three features form a threat model. The threat model is an aggregation of security aspects to address in the system. The threat model is needed to formulate a set of security objectives, describing the security features and services we need to have in the solution.

8.3. Distributed Control Systems

A DCS refers to a control system, in which the controller elements are not central in location, but distributed throughout the system, and each subsystem is controlled by one or more collaborating control units. The entire system of subsystems and controllers is connected by a network for communication and monitoring. DCS is a very broad term used in a variety of areas, to monitor and control distributed equipment. However, for any application based on DCS to be fully functional, information must be interchanged in a timely and reliable manner among the controllers, actuators and sensors in the communication network. Thus, DCS has stringent requirements on both reliability, availability and timeliness, the latter in terms of real-time deadlines. If the deadline is missed, the packet is considered useless, similarly to a lost or erroneous packet in a non-real-time system. The effect of packet scheduling attacks on shipboard networked control systems was evaluated in [8] and clearly shows that such attacks can be easily mounted to both wired and wireless communication channels leading to time varying delays packet scheduling anomalies.

Traditionally, most DCSs are based on wired networks, e.g. fieldbus systems [9] such as the CAN bus, HART, FlexRay or PROFIBUS. However, employing wireless communications offers significant advantages to DCSs, as they become mobile, autonomous and connected [3, 10, 11]. Wireless channels are, by nature, more exposed to noise and interference than their wired counterparts. Consequently, it implies a considerable challenge to fulfill the real-time requirements with sufficient reliability for proper functionality of applications based on DCSs [3, 11]. In particular since for wireless DCS, not only system reliability needs to be considered, but also individual link reliabilities.

Since the use of wireless communications in DCS becomes more and more common and since it is difficult to solve all application requirements with only one communication standard [2, 12], heterogeneous networks are introduced. By heterogeneity is meant that the networks consist of different types of nodes and different communication links. Most emerging DCS also needs to support several types of data traffic classes with different levels of criticality. This is termed a converged network, typically supporting three different traffic classes: Time-Triggered (TT), Even-Driven (ED), and Best-Effort (BE) traffic. These three traffic classes have different applicability, scheduling requirements and temporal characteristics.

It is rarely possible to add a security framework to an existing system without considerable changes [12]. For example, time-triggered systems, frequently encountered in DCSs, imply the need for scheduling. Scheduling is an essential part of the correctness of the system. Security implementations on top of such a system can affect the scheduling, impose delays and increase the frame length [13], and, therefore, security objectives should be considered in the scheduler as well. Obviously, if the security constraints are taken into account already at the system design stage, it has several advantages.

8.4. System Model

Note that we do not target a specific use case, but a wide range of use cases based on DCS. In order to specify and characterize them, their common aspects related to distributed control are considered. Hence, the use case is systems requiring high reliability, timeliness and availability and where something can be controlled or interfered with from a distance if security is breached. We target heterogeneous networks consisting of different types of nodes and communication links with support for three different data traffic classes TT, ED and BE. Also, to be able to consider time-critical applications, security support for applications with real-time constraints needs to be added as a requirement, as security solutions always affect the delay and time properties of the system. Hence, synchronization is an important asset.

A. Security objectives and Threats

To characterize the required security profile for DCSs, we need to derive the adversary goals, the system assets and the adversary model [14] for it as it proposed in Fig. 8.1. This will allow us to build a threat model for the profile. Based on the threat model we can derive the profile security objectives. The security objectives specify what types of threats the system is to be secured against.

For the use case communications in a DCS to be feasible, it must be safe for humans and the environment, provide timely and reliable data delivery and be available to perform the task at hand. As we consider applications with real-time requirements, one of the main system assets is clock synchronization and thus many attacks target clock synchronization [15]. For some

DCSs carrying sensitive information, data confidentiality can be considered an asset. In this case a potential adversary can exploit eavesdropping to achieve his goal. However, for many DCS, it does not matter if data can be eavesdropped as long as it cannot be compromised or altered. A particular feature of DCS in this respect is that in order to compromise the data, it is enough to delay it, since in a real-time system it is not only the data itself that is of importance, but also when in time it is presented. As TT traffic requires a low jitter, it is enough for an intruder to cause a random delay of TT packets affecting the periodicity, to compromise the functionality. For ED traffic, a temporal delay of an alarm or a warning, will cause system malfunction. A delay therefore also affects the reliability and availability of the system. For safety-critical systems, availability is one of the main assets, as system shutdown has an extremely high cost. A Denial of Service (DoS) attack can then threaten to decrease the availability of the system. Certain safety features often implemented in DCSs for increased data reliability, like robustness and fault tolerance mechanisms, can directly contribute also to achieving specific security goals, such as excluding the possibility of DoS attacks.

There are many possible goals for an adversary targeting DCSs, but according to [16] the following three can be considered as general adversary goals: system disruption, eavesdropping and system hijacking. All three are possible within the considered profile and will lead to that the adversary can control or interfere with something from a distance if security is breached. If we consider a power plant generating electric power to a city, the adversary can try to disrupt the system. For factories producing products according to secret recipes, like Coca-Cola, the adversary is likely to target system eavesdropping. Even machines that operate in a closed system without contact to the outside world can be eavesdropped upon via monitoring electro-magnetic transmissions generated by the hardware. In a sensor network checking paper humidity during production, an adversary can target system hijacking, so that he can replace the sent data during its transmission. In this case, an incorrect function of the paper machine can be hidden from the control center for a long time, causing tremendous money loss for the factory. One important topic in the near future is hijacking, since communication links, not properly protected, are appealing for an adversary. This is especially important for DCS, as an adversary can e.g., target automated factory hijacking or vehicle control hijacking which can lead to significant damages or even loss of human lives.

The adversary model strongly depends on the specific use case, but we can still list several generally applicable suggestions about the adversary. We assume that he is capable of intercepting any transmitted message and he is also able to alter and retransmit it. In addition he can create his own messages and perform data processing using publicly available system credentials. Besides these abilities, the adversary can also be characterized by the following criteria: whether he is adaptive or static, his computational power (typically proportional to the targeted systems assets and values), whether he is mobile, etc.

8.5. Evaluation Methodology

In this paper, we consider a security framework to be a set of interconnected solutions that satisfy the system requirements on security by providing all necessary security services required within the network. To investigate existing security solutions and frameworks, we need to have a tool to evaluate them. The set of security objectives derived from our targeted use case can be such a tool, as we can compare solutions through the subsets of security objectives they are able to cover.

A. Security Concepts

An attack is an intentional attempt to break a security objective made by an adversary. There are innumerable types of attacks for all kinds of applications, and the amount grows every day. Nevertheless, it is possible to provide a general classification of possible attacks types relevant to most types of DCS. According to [5], attacks can be grouped into the following categories: DoS, eavesdropping, man-in-the-middle (when an attacker pretends to be a legitimate partner in the middle of the communication process) and intrusions (breaking into a system using a virus, a Trojan, or a worm, i.e. a malicious code that targets to exploit system vulnerabilities). All attacks target to violate the security objectives of the system. The classical triad of security objectives is Confidentiality, Integrity and Availability (CIA). Considering the DCS scope it can be completed with authentication, authorization, non-repudiation, and third party protection [17]. For each use, case it is possible to form a list of objectives that are specific for that particular application.

B. Security for wired and wireless networks

Traditionally, security has not been considered from the very beginning in wired networks for DCSs, as they conventionally are both static and closed. Mobility is seldom required, so the network configuration is usually static. In wired networks it is more difficult to join the network illegally, compared to a wireless one, and, therefore, many wired networks are regarded as closed.

Moreover, some wired networks are considered to be secure due to their location, especially in environments, which are difficult to reach, e.g. the embedded systems in cars. Therefore, the probability to breach the system in such networks is considered low, but, of course, everything depends on the gain and the system values. However, with growing levels of automation in diverse application areas, security concerns in wired industrial networks are rising as well.

In contrast, security, especially in terms of encryption, is basically always considered for wireless networks. The security approaches for wireless networks reflect specific features such as easy access to the communication links, vulnerability due to the propagation characteristics of wireless signals etc. Different approaches imply different tradeoffs between security services and network performance [18].

C. Security solutions on different layers

Security solutions can be implemented on different layers of the OSI model. Usually, the application, network and link layers are considered as main candidates for introducing security features. In this subsection, all three options are investigated and compared from the specific point of view of developing a flexible security framework for heterogeneous networks.

Security features implemented on the application layer can allow end-to-end checking of the system assets and also provides a high security level due to targeting a specific use case. For security on the lower layers in the communication stack, a more general approach needs to be adopted, whereas application layer security solutions are a good choice when the application can be well specified. One example of such a solution is presented in [19], where the author proposes a security solution targeting initial trust establishment for industrial heterogeneous networks. On the other hand, security solutions on the application layer can lead to lack of flexibility and also a factor, which needs to be considered, is that the user interacts directly with the security solution and thus the security performance can depend on the individual user's skill.

The network layer is widely studied for applying security solutions. A logical conclusion from some of these studies was the development of IPSec [20]. The main advantage of this approach is that security solutions implemented here will be hidden from the user. Moreover, it is suitable for heterogeneous networks without any precise connection to the specific use case being necessary nor exact knowledge about the underlying links.

Security solutions can also be implemented on the link layer. One example of an architectural framework for security on the link layer was presented in [21]. The authors propose a security solution over Ethernet, providing such services as key management and countermeasures to DoS attacks. Generally, the link layer is a convenient place for data management security solutions, dealing with bootstrapping, access control etc. The main disadvantage is dependency on system configuration, and knowledge about whether it is a wireless or wired network, as these have different mechanisms for channel access (MAC sub-layer) and services provision (LLC sub-layer).

A security framework implementation can also be done as a separate additional layer, it can be merged with an existing layer or it can be split between layers, so that different layers have different security functionalities. The latter is often more efficient, as it allows combining the advantages of security implementations on several different layers.

D. Security Framework for DCSs

The security framework should be designed considering the exact set of security objectives derived from the use case. Such an approach allows efficient and adequate protection of the system assets. Also, the suitability of security framework depends on the specific system structure. For heterogeneous DCSs, it should include solutions to achieve an appropriate level of safety in the wireless part of the system and a corresponding level of security in the wired part of the system [17].

The security framework can be implemented on different layers in the OSI model, as each layer has its own advantages and disadvantages. Therefore, the most promising solution seems to be a cross-layer architecture. For example, the core solution can be implemented in one layer, and completed by a set of optional extensions operating on others layers.

8.6. Evaluation of Existing Security Solutions

In this section we investigate existing security solutions specific for either wired or wireless networks. The majority of the existing solutions evaluated in this paper comes from the area of industrial control. The reason for this is that this is the most mature application area exploiting DCS and thus all protocols and security solutions are already in commercial use since several years. However, most of them do not support emerging applications based on DCS, becoming mobile, autonomous and connected.

Different solutions are available for wired and wireless systems, and solutions can also vary depending on where in the OSI model they are applied and what type of traffic policy is used in the network.

While adversaries constantly come up with new types of attacks, system developers continue to design new countermeasures. Countermeasures as well as security solutions can specifically target wired or wireless networks, and they can be applied on different levels in the OSI stacks resulting in different properties. On the physical layer, a possible countermeasure to channel jamming is frequency hopping. On the application layer, a possible countermeasure to eavesdropping is encryption.

A. Solutions for Wired Sensor Networks

DCS are traditionally based on wired networks. Most of these have weak support for security that demonstrates the need for a flexible security framework, able to meet the requirements of merging application areas with high demand on reliability, timeliness and availability.

HART (Highway Addressable Remote Transducer) is an industrial automation protocol, which is widely used in factory automation as a reliable and long-term solution for plant operators. However, considering security aspects it only has single parity checking [22]. Single parity check codes, originally intended to detect communication errors caused by the channel, can also be used as an indication that part of the message was intentionally changed. Hence, in this sense it is a security solution, albeit not a strong one, as it cannot help when a malicious node changes the message and recalculates the checksum. HART is thus an example of a network that traditionally has been considered closed and thereby better protected.

TTEthernet (Time-Triggered Ethernet) [23] is a platform that extends classical Ethernet so that it becomes possible to use for safety-critical application with real-time requirements. This protocol can be used for systems that have several levels of time and safety requirements due to its support for several different traffic classes. However, like HART, TTEthernet does not provide any specific security service [24].

CAN (Controller Area Network) is one of the most broadly used technologies for in-vehicle communication. Regarding security properties, CAN supports data transfer security, i.e. it can detect an error and signal about it. Its main security weaknesses were investigated in [25], and they include initial broadcast nature of all packets, extreme vulnerability to DoS attacks, lack of authentication fields, and week access control. All these weaknesses originate from the initial assumption that networks using the CAN protocol are closed for intruders, but it is not always a case. Especially, when a wireless gateway is introduced.

Evaluation. Neither CAN nor TTEthernet support heterogeneous networks and neither CAN nor HART can provide real-time support for three different types of data traffics. It is also noticeable that the set of security objectives for each solution depends on the concrete application area.

B. Solutions for Wireless Sensor Networks

There are several technologies and standards that are intended for wireless sensor networks and many researches target their future exploration, comparison and development. IEEE 802.11 (WiFi), IEEE 802.15.1 (Bluetooth) and IEEE 802,15.4 (ZigBee, WirelessHART) are compared and investigated in [26] as possible solutions for sensor networks in industrial applications. WirelessHART is mentioned there, as one of the most robust protocols with high quality real-time performance technology, but having poor link throughput and network scalability.

Also, there are some existing researches evaluating security aspects in wireless sensor networks. IEEE 802.15.1, IEEE 802.11 and IEEE 802.16 (WiMAX) are compared in [27] from the security point of view for the use case of wireless sensor networks. The author considered such security techniques as authentication, key-distribution and cryptographic concept, their specific features and flaws. In all these standards and technologies, different security approaches and techniques are used. Some examples are considered in more detail below.

An analysis of the security support within the 802.15.4 standard was made in [28]. A link layer security protocol provides the following services: secure access control, message integrity, message confidentiality and replay protection. Also there are eight different security suits that imply various combinations of security services and three supporting keying models. The authors outline such protocol weaknesses as lack of group keying, easy to break integrity protection etc.

Evaluation. The technologies mentioned above do provide security services, but suffer from the lack of safety. Therefore, to be able to cover heterogeneous networks they need to be complemented with suitable safety extensions. Lack of reliability leads to straitened TT data traffic support and reduced availability. Safety and security protocols are similar, as they are designed to limit the probability of malfunction or misuse. The main difference is that in order to reduce the risk of non-authorized access or similar, security protocols use cryptography, instead of relying on CRCs only, as most safety protocols do. A combination of these methodologies, limits the probability of successful attacks further, as safety protocols will detect deviations in timing or in configuration data immediately, which is yet another barrier for an intruder to bypass in order to get unauthorized access. If the safety protocol detects runtime deviations, the system will go to a safe state mode and can only be put into operation again by manual and a physical reset of the safety system. In addition, the safety system will alert the staff by high priority alarms when a deviation is detected for sake of limiting harm to people, property or environment.

8.7. Evaluation of Existing Security Frameworks

We consider here several examples of security frameworks representing different approaches of development; a framework for mixed wired and wireless networks and frameworks applying security on the application and IP layers respectively.

A. Security framework for heterogeneious networks

In [17] the author presents a framework for mixed wired and wireless industrial sensor networks using HART and WirelessHART for communication in process automation. It can be concluded that there is usually a lack of security in wired networks and a lack of safety in wireless networks, as traditionally most systems have only one of them [29]. The author targets bringing both security and safety together for heterogeneous networks. The solutions are retrofitted on an existing architecture, to enable integration of wireless communications into existing wired networks. The idea is based on introducing a security module that provides end-to-end security and maintains authentication, integrity and confidentiality. This module is an additional security layer used on top of PROFINET IO [30]. PROFINET IO deals with peripheral devices and controls data-exchange. The framework treats safety and security in the same way and hides the differences. It is based on the idea of the black channel with which each level in the network provides services for both safety and security without relying on other layers. The black channel principle is based on that the safety layer has to implement measures to all possible error cases, and not rely on existing measures in other layers, in order to avoid a safety case for all intermediate layers, components and nodes. Thus, this introduces redundancy since most of the error cases are handled in parallel by other layers as well, but with varying residual risk of detecting each error case. The security and safety layers are thus separated and can be deployed independently. Such an approach can be considered as an example of the "defense-in-depth" method.

Evaluation. The framework supports mixed wired and wireless networks and its wireless part covers such security objectives as authentication, integrity and confidentiality, which is a benefit. However, the framework does not support ED or BE traffic classes. This solution is used in industry for many applications, but its applicability for highly critical use cases still needs further evaluation.

B. Security framework on the application layer

One example of a security framework working on the application layer is presented in [19]. The framework includes iterations with users in the loop, and solutions embedded into devices, hidden from the users are mostly discussed. The author proposes an approach for initial trust bootstrapping and life cycle managing. The application area is industrial networks and the solution is valid for heterogeneous networks. The security objectives considered are availability, device authentication, confidentiality, and system resilience. The proposed framework is based on the workflow that regulates the communication between the users of the network and the network itself. The framework consists of three main phases: initial trust establishment, device verification, and key generation. The approach was proved by implementation. The workflow protocol supports predefined configuration of the involved parameters (an approach which is common for wired networks), as well as adaptable configuration, when a device can join the network after successful authorization (an approach common in wireless networks). Also the framework supports the possibility to enter configuration data manually in new devices. Finally, it is able to support symmetric and asymmetric key distribution and does not require using shared predefined secret parameters. Another advantage of the solution is that the framework is flexible considering dynamic roles of employees.

Evaluation. This is a high-level framework. Therefore, it is suitable for heterogeneous networks and for different traffic-classes as there are no dependencies with lower layers. This independence is one of the main advantages. However, this also leads to a more limited set of possible security objectives. The framework targets authorized network access, but neglects such objectives as availability and reliability. This solution can be a part of a many-layers framework for DCS. It can be combined with low-level solutions that would protect, e.g., clock synchronization.

C. Security framework on the network layer

IPSec (Internet Protocol Security) [31] operates on the network layer and supports such security services as message integrity authentication, data encryption by different encryption algorithms, and message replay attack protection. There are two feasible ways of its realization, either in the End-Host or in the Router. Also, there are three possible architectures: integrated, BITS (bump in the stack) and BITW (bump in the wire). The first one implements IPSec directly into the IP layer itself. This way looks very natural, but since only IPv6 was designed to support IPSec, IPv4 still have to use BITS. BITS implies that IPSec becomes a separate layer between the network (IP) and the data link (network interface) layers, which will introduce some overhead in the stack. Finally, BITW assumes that hardware devices that can provide IPSec services can be added, which may or may not be possible. Authentication Header (AH) and Encapsulating Security Payload (ESP) protocols are two principal parts of IPSec. AH supports message integrity, data authentication and replay attack protection services, whereas ESP supports encryption of whole datagrams. IPSec deals with keys through the Internet Key Exchange (IKE) protocol.

Evaluation. The framework is suitable for heterogeneous networks and can potentially support different data traffic classes. However, it lacks real-time support, but this could potentially be achieved by using an extension targeting enhancement of real-time properties, e.g., as in [14]. The framework also covers such basic security objectives as data integrity, confidentiality, and authentication. However, the ability to provide objectives such as reliability and availability will depend on the concrete implementation as well as the application requirements.

D. Frameworks comparison

The three considered frameworks are examples of solutions deployed on different layers and targeting different security objectives. Nevertheless, they can still be compared based on suitability for DCS. In Table 8.1 columns A, B, and C stand for the corresponding frameworks from subsections A, B and C respectively. CIA in the table stands for confidentiality, integrity and authentication. The evaluation shows that if we target several security objectives in one system, none of the frameworks fulfill all requirements. This means that existing frameworks must be enhanced and combined in order to achieve the desired security level in networks for DCSs.

Table 8.1 Identifier Declaration

Feature	A	B	C
Mixed network support	✓	✓	✓
Different traffic classes support	✗	✓	✗
Real-time support	✓	✓	✗
Covered security objectives	CIA for wireless part	Authorized network access	CIA

8.8. Conclusions and Future Work

In this paper, we have investigated existing security solutions and frameworks suitable for a distributed control systems profile. We also specified the security requirements for the considered application domain and investigated the basic component of threat modeling. We considered three main types of classification of security frameworks; depending on if wired or wireless communication links are included, if support for different data traffic classes is provided and, finally, the layer in which the security framework is implemented on. We also investigated several existing security solutions and frameworks for wired and wireless industrial sensor networks and found that although they all have different benefits and drawbacks, and no existing solution or framework has all the required properties. We can also conclude that if security can be added already at the design stage, much is gained.

Acknowledgments

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n°607727. E. Uhlemann is partly funded by the Knowledge Foundation through the WIRE project. E. Uhlemann, J. Åkerberg and M. Björkman are partly funded by the Knowledge Foundation through the READY project.

Bibliography

[1] P. Gaj, J. Jasperneite, and M. Felser, "Computer Communication Within Industrial Distributed Environment—a Survey," IEEE TII, vol. 9, no. 1, pp. 182-189, Jul 2012.

[2] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in Proc. INDIN, Portugal, July, 2011, pp. 410-415.

[3] V.C. Gundor and G.P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," IEEE TIE, vol. 56, no. 10, pp. 4258-4265, 2009.

[4] M. Cheminod, L. Durante, and A. Valenzano, "Review of Security Issues in Industrial Networks," IEEE TII, vol. 9, no. 1, pp. 277-293, Feb 2013.

[5] D. Dzung, M. Naedele, T. P.Von Hoff, and M. Crevatin, "Security for Industrial Communication Systems," Proc. IEEE, vol. 93, no. 6, pp. 1152-1177, 2005.

[6] H. Kopetz and W. Ochsenreiter, "Clock Synchronization in Distributed Real-Time Systems," IEEE Transactions on Computers, vol. C-36, no. 8, pp. 933-940, 1987.

[7] M. Ullmann and M. Vögeler, "Delay attacks - implication on NTP and PTP time synchronization," in Proc. International Symposium on Clock Synchronization for Measurenment, Control and Communication (ISPCS), Brescia, Italy, Oct., 2009.

[8] E. Penera and D. Chasaki, "Packet scheduling attacks on shipboard networked control systems," in Proc. Resilienca Week (RWS), Philadelphia, PA, 18-20 Aug., 2015, pp. 1-6.

[9] J.-D. Decotignie, "Ethernet-Based Real-Time and Industrial Communications," Proc. IEEE, vol. 93, no. 6, pp. 1102-1117, Jun 2005.

[10] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Networks," Proc. IEEE, vol. 93, no. 6, pp. 1130-1151, 2005.

[11] A. Willig, "Recent and Emerging Topics in Wireless Industrial Communications: A Selection," IEEE TII, vol. 4, no. 2, May 2008.

[12] J. Åkerberg, M. Gidlund, T. Lennvall, J. Neander, and M. Bjorkman, "Efficient Integration of Secure and Safety Critical Industrial Wireless Sensor Networks," EURASIP J. on Wireless Commun. and Netw., vol. 2011, no. 1, November 2011.

[13] T. Xie and X. Qin, "Scheduling security-critical real-time applications on clusters," IEEE Transactions on Computers, vol. 55, no. 7, pp. 864-879, Jul 2006.

[14] E. Lisova, E. Uhlemann, J. Åkerberg, and M. Björkman, "Towards secure wireless TTEthernet for industrial process automation applications," in Proc. Emerging Technologies and Factory Automation (ETFA), Barcelona, Spain, Sep., 2014.

[15] T. Mizrahi, "Time Synchronization Security using IPsec and MACsec," in Proc. IEEE ISPCS, Munich, Germany, Sep, 2011, pp. 38-43.

[16] M. Anand, E. Cronin, M. Sherr, M. Blaze, Z. Ives, and I. Lee, "Sensor network security: more interesting than you think," in Proc. USENIX HotSec, San Jose, CA, Jul, 2006, pp. 5-5.

[17] J. Åkerberg, On Safe and Secure Sommunication in Process Automation, Doctoral thesis, Mälardalen University, 2011.

[18] R. A Gupta, A. K. Agarwal, M.-Y. Chow, and W. Wang, "Performance Assessment of Data and Time-Sensitive Wireless Distributed Networked-Control-Systems in Presence of Information Security," in Proc. MILCOM, Orlando, FL, Oct, 2007, pp. 1-7.

[19] A. Ray, Initial Trust Establishment For Heterogeneous Industrial Communication Networks, Licentiate thesis, Mälardalen University, 2014.

[20] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 24011998.

[21] H. Altunbasak and H. Owen, "An arhitectural framework for data link layer security with security interlayering," in Proc. SoutheastCon, Richmond, VA2007, pp. 607-614.

[22] S. Raza, A. Slabbert, T. Voigt, and K. Landernas, "Security considerations for the WirelessHart protocol," in Proc. ETFA, Mallorca, Spain, Sep, 2009, pp. 1-8.

[23] W. Steiner, G. Bauer, B. Hall, and M. Paulitsch, "TTEthernet: Time-Triggered Ethernet," in Time-Triggered Communication, R. Obermaisser, Ed., ed CRC Press, IDate, 2011.

[24] W. Steiner, "Candidate security solutions for TTEtnernet," in Proc. DASC, East Syracuse, NY, Oct, 2013, pp. 1-10.

[25] K. Koscher, A. Czeskis, F. Roener, and S. Patel, "Experimental Security Analysis of a Modern Automobile," in Proc. IEEE Sym. SP, Oakland, CA, May, 2010, pp. 447 - 462.

[26] S. Giannouslis, C. Koulamas, C. Emmanouilidis, P. Pistofidis, and D. Karampatzakis, "Wireless Sensor Network Technologies for Condition Monitoring of Industrial Assets," in Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services, ed: Springer Berlin Heidelberg, IDate, 2011, pp. 33-40.

[27] G. Lackner, "A Comparison of Security in Wireless Network Standards with a Focus on Bluetooth, WiFi and WiMAX," Int. J. Netw. S., vol. 15, no. 6, pp. 420-436, 2013.

[28] N. Sastry and D. Wagner, "Security Considerations for IEEE 802.15.4 Networks," in Proc. ACM WiSe, Philadelphia, Pennsylvania, Oct, 2004, pp. 32-42.

[29] C. W. Axelrod, "Applying Lessons from Safety-Critical Systems to Security-Critical Software," in Proc. IEEE LISAT, Farmingdale, NY, May, 2011, pp. 1-6.

[30] A. Poschmann and P. Neumann, "Architecture and Model of Profinet IO," in Proc. AFRICON, Sep, 2004, pp. 1213-1218.

[31] S. Kent and S. Seo, Security Architecture for IP IETF RFC 4301, Dec 2005, http://www.rfc-editor.org/rfc/pdfrfc/rfc4301.txt.pdf

Chapter 9

Paper C: Risk Evaluation of an ARP Poisoning Attack on Clock Synchronization for Industrial Applications

Elena Lisova, Elisabeth Uhlemann, Wilfried Steiner, Johan Åkerberg, and Mats Björkman.

In Proceedings of IEEE International Conference on Industrial Technology (ICIT*)*, Taipei, Taiwan, March 2016.

Abstract

Nowadays, mixed wireless and wired networks are used everywhere in everyday life, including in industry where they often support time-critical applications. Industrial applications with high precision requirements are subject to real-time constraints, and thus one of the main assets, regardless of application area, is clock synchronization. Considering such networks, synchronization is the first thing to secure against a possible malicious adversary. In this paper, we consider ARP poisoning as a possible technique to disrupt clock synchronization and evaluate the effects of such an attack on the IEEE 1588 standard. We describe possible ways of performing ARP poisoning to disrupt synchronization and survey several mitigation techniques and their applicability within the industrial application area.

9.1. Introduction

The application field of clock synchronization algorithms is very wide. It includes all networks with a time-triggered architecture, i.e., networks where message transmissions should be made within time-slots assigned according to an offline or online schedule. An excellent example is mixed wired and wireless industrial networks that are allocating slots for exchange of time-critical messages [1]. Here, real-time properties and predictable delays are essential to ensure full system availability even for critical applications. Depending on the concrete application area, the possible consequences and costs of breaking clock synchronization vary. We target industrial applications, where the prize of failure is high and where clock synchronization can be considered as one of the main system assets [2]. If a node is unsynchronized, it cannot communicate correctly with the other network participants due to the real-time properties and the requirement on time-slot synchronization, and thus the system loses in reliability and availability. If an intruder wants to disrupt the network, there is no need to determine the assets of the specific application or analyze all algorithms and protocols used in the networks to determine their weak spots – it is enough to influence the clock synchronization algorithm. Consequently, protecting clock synchronization becomes a prime issue in industrial networks.

A commonly used standard for clock synchronization is IEEE 1588, the Precision Clock Synchronization Protocol [3]. It contains a network protocol responsible for precise synchronization of heterogeneous systems nodes that can have clocks with different parameters regarding stability, resolution etc., and provides a precision better than one nanosecond. The revised standard from 2008 has an optional secure extension called Annex K [3], which provides group source authentication, message integrity and replay attack protection. We consider IEEE 1588 in our investigation of the possibilities and implications of breaching clock synchronization, as this standard is widely used in the industrial communication area.

In order to protect clock synchronization in the system, we need to investigate how a possible intruder can breach it. There are several studies investigating the weak spots of IEEE 1588 in general, e.g., introducing artificial delays [4], but our work targets a possibility which has not been evaluated previously: namely to breach clock synchronization using a well-known technique called Address Resolution Protocol (ARP) poisoning. ARP is a protocol used to define the correlation between network and link layer addresses, i.e., it establishes the accordance between the Internet Protocol (IP) addresses and the Medium Access Control (MAC) addresses. An ARP poisoning attack implies that by using ARP protocol loopholes, such as lack of authentication, it is possible to perform a man-in-the-middle attack on the network [5]. As a result of an attack, two targeted nodes will think that they are exchanging messages with each other, but in reality they will communicate through the intruder. The ARP protocol is commonly used in many networks, often without any protection techniques, and thus the ARP poisoning attack is well known and there are many research studies investigating its countermeasures, e.g., [5-7].

Consequently, two facts are well known: the possibility to break the synchronization in IEEE 1588 by imposing artificial delays and that ARP poisoning can be used to gain control over the communication channel between different network participants. Therefore, the combination of these two facts together, using an ARP poisoning attack to impose a delay in order to break synchronization, leads to the possibility to influence many types of industrial networks via one single type of attack. In this paper, we provide a detailed investigation of the possibility to use ARP poisoning to breach clock synchronization in networks that use IEEE 1588. The proposed approach is evaluated in the security protocol simulator AVISPA. In addition, possible mitigation and protection techniques are investigated for applicability in the industrial application area. Given the targeted application area, several restrictions are imposed on the solutions, as most of them add additional communication delays, which can be critical for time-triggered applications.

The remainder of the paper is organized as follows. Section 9.2 presents the system model and requirements for security solutions targeting the industrial application area. In Section 9.3 we give an overview of the IEEE 1588 standard, its weak spots and its security solutions, whereas a vulnerability analysis is done in Section 9.4. The ARP poisoning attack is discussed in Section 9.5. In Section 9.6, the attack analysis, its applicability and consequences are investigated. Section 9.7 presents an overview of possible mitigation techniques for ARP poisoning. An evaluation of the approach and our results are described in Section 9.8. Finally, Section 9.9 provides the conclusions and description of future work.

9.2. The System Model and Security Requirements

Today's market is looking towards wireless solutions, which besides a set of obvious benefits, also introduce many security issues, since wireless links are more open for malicious intruders. Consequently, security aspects are becoming more and more important for industrial networks [8]. We target heterogeneous networks, i.e., a mix of wireless and wired links, organized in a mesh topology, consisting of nodes, switches and access points for the wireless parts of the networks. Note that some simple network topologies encountered in industrial settings, such as the star network, are not considered in this paper, as they can be implemented using the physical, link and application layers only. In such networks, the need for network addresses is limited, and thus ARP may not be needed.

There are many challenges in securing industrial networks [8], such as limited recourses, long lifetime of equipment, third party support etc. Networks can be diverse depending on the application, but the majority of all industrial networks have clock synchronization as one of the main system assets. Therefore, such an attack can be applied as a universal system disruption technique independent of the specific use case. This fact makes the attack and its countermeasures important since if the same technique can be applied in many different cases it is more likely that the adversary will invest resources in it.

For security solutions to be beneficial in industrial settings, the following aspects should be considered. Every solution implies some additional communication overhead, and for networks with low latency and high throughput requirements, this can be critical. Also industrial networks can be used for safety critical applications and from that perspective it is important that a possible solution does not introduce single points of failure, as this can reduce the availability. Another important requirement for industrial networks is backwards compatibility. There is a tremendous amount of equipment already installed and in order to be implementable in practice, a proposed solution cannot require replacement of all existing equipment.

9.3. Clock Synchronization and IEEE 1588 Standard

In this section we give an overview of clock synchronization issues, the IEEE 1588 standard and its specific features.

A. IEEE 1588 standard

IEEE 1588 is a commonly used standard for clock synchronization in industrial applications, such as, for example, substation automation [9]. This standard includes the Precision Time Protocol (PTP), which implies a master-slave approach for handling the synchronization. The approach allows the network to be self-organizing. At any point in time, the network can choose and assign the clock with the highest precision to which all other clocks must be adjusted.

The correction of clocks according to the current grandmaster clock is done via exchange of synchronization messages. During this exchange, the slave clock calculates its drift and as a result performs a correction. This approach is based on two assumptions:

- A message needs the same time for being transferred from node A to node B as for being transferring in the opposite direction, from node B to node A, i.e., that the delays are the same in both directions.
- The messages exchange can be done in short enough time so that the information acquired about the clock drift is still valid and can be used for correction, i.e., that the calculated offset can be considered correct.

B. Security analysis of IEEE 1588

A classification of possible threats and security breaches for the IEEE 1588 standard from a digital substation automation point of view was presented [9]. The authors propose to classify the possible attacks into five categories: network/processing queue congestion; removal of messages; selective packet delay; packet modification; and masquerading as master. However, the digital substation automation network is more isolated than general industrial networks, and therefore the solutions proposed in the paper are not directly applicable. A threat analysis for IEEE 1588 was presented in [4], the considering the security objectives integrity, authentication and availability. Also the authors proposed to divide all possible attacks into the following main groups: direct attacks on a node, byzantine masters, message manipulation and message delay and insertion. In [10] the authors investigated PTP and the Network Time Protocol (NTP) reaction to delay attacks and its consequences. For the PTP protocol, two approaches were investigated: message delaying and acceleration. In our work, both scenarios are possible, but we select message delaying as it is easier to perform. Based on [4, 9, 10], we can conclude that selective packet delaying is a threat to the asset clock synchronization, as it will cause reduced reliability and availability of the system.

C. IEEE 1588 Security Extensions – Annex K

The second version of IEEE 1588 from 2008 has an optional security extension called Annex K. The proposed security measures can be divided into two groups. The first group deals with message integrity protection and the second one provides guidelines for group source authenticity.

An analysis of IEEE 1588 Annex K was conducted in [11]. The authors conclude that the proposed technique for message integrity correction is suboptimal, as the sequence number introduced to tolerate replay attack is too short and the proposed three hand-shake authentication procedure can be simplified to a one-shake procedure. These results show that the security extensions of 1588 need to be developed further, as there are still a number of open security issues and also the addressed solutions can be enhanced further. In [9] the authors investigate how Annex K can help against the five categories of attacks targeting IEEE 1588 mentioned above. Attacks from the categories packet modification and masquerading as master can be prevented by the security solutions proposed in Annex K. Also it partly helps against network/queue congestion, but it cannot help against removal of messages or selective packet delay. The last one is important for our work, since selective packet delay can be used to break clock synchronization without being detected. Messages authentication cannot help against this type of attacks, as an intruder does not need to change the message, just delay it. The same goes for replay attack prevention measures. Source authentication cannot help either, as we consider a delay imposed via ARP poisoning, the technique that enables hijacking and controlling of the entire communication channel. This shows that there is a need for an additional technique to protect clock synchronization from selective delay attacks.

9.4. Vulnerability Analysis

The general idea of clock synchronization in a network is that clock corrections are done periodically and that due to this correction, the clock drift does not exceed the maximum upper bound allowed within the period duration. Independently of which specific synchronization algorithm that is responsible for data accumulation and decision-making, we can calculate how much the clock needs to be shifted at the correction points in order to breach the synchronization during a specific period. Ideally, a node's clock shows time t_{clock} that linearly, with a constant coefficient 1, depends on real time t_{real} (the line with two dots and a dash in Fig. 9.1). In reality, however, we cannot guarantee this in any distributed system, and hence it is required that the green line in Fig. 9.1, showing the behavior of t_{clock} with respect to t_{real} is within the real time plus, minus $\varDelta t_{max}$. The clock correction procedure is executed every $\varDelta t_{per}$, and the value of $\varDelta t_{per}$ depends on the clock drifts. Therefore, to be synchronized, the line showing the real dependency (solid green line), is shifted in steps every $\varDelta t_{per}$, such that it stays within the bounds.

If we have a system with only two clocks A and B with t_{clockA} and t_{clockB}, they are considered to be synchronized with the precision Δt_{max} if at any point of time, the difference between their local times is less than Δt_{max}. In other words, as one of the clocks A or B will be selected to be the current grandmaster clock, its time will be interpreted as t_{real} and thus we are only interested in the difference between the two clocks. Therefore, the condition for breaching synchronization is:

$$\left| t_{clockA} - t_{clockB} \right| > \Delta t_{max} \tag{9.1}$$

According to IEEE 1588 [1], a slave clock calculates its offset to the master, Δt_{offM} as:

$$\Delta t_{offM} = t_{inS} - t_{outM} - t_{corr} \, , \tag{9.2}$$

where t_{inS} is the time of arrival of the synchronization message to the slave; t_{outM} the time when the synchronization message left the master, and finally t_{corr} a correction variable, which incorporates the propagation delay of the message. Hence, if the slave clock is completely synchronized to the master clock, $t_{inS} - t_{outM}$ would be equal to the propagation delay and thus $\Delta t_{offM} = 0$, and the slave clock would be left unchanged. If an adversary influences only this particular synchronization message such that t_{inS} is increased, a new asynchronous delay that cannot be compensated by the algorithm would be introduced and the slave clock would be adjusted to a fake clock.

If the imposed delay is bigger than Δt_{max}, it would lead to fulfillment of the synchronization breaching condition (9.1). It is interesting to note that the adversary can make the slave clock slower or faster depending on which message in the message exchange process is delayed. The sign of delay depends on in which direction (master-slave or slave-master) the message is delayed.

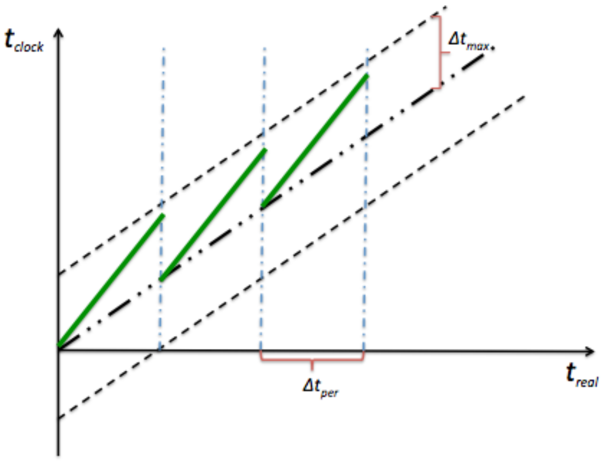

Fig. 9.1 Node clock drift, blue line – the actual time of the clock that should stay within the bounds (dotted lines) in order to keep the node synchronized.

In order to compensate for clock drift all slaves shift their clocks periodically at time instance $\{i, i{+}1, \ldots\}$ according to:

$$t_{clock}^{i+1} = t_{clock}^{i} - \Delta t_{offM} \ . \tag{9.4}$$

The clock drift is a random value affecting the angle of the green line in Fig. 9.1. Generally, the more expensive a clock is, the lower clock drift is guaranteed by its manufacturer. Let us define slave clock time as:

$$t_{clock} = \alpha \cdot t_{real} \ , \tag{9.5}$$

where α is the clock drift: if $\alpha < 1$, the clock is defined as fast; and if $\alpha > 1$, the clock is defined as slow. Given that α is a random value, it is not enough that the delay imposed by the adversary is bigger than Δt_{max} if the clock is fast. In the worse case, when the clock is as fast as it is possible given allowed bounds, the imposed delay must be more than $2\Delta t_{max}$. Conversely, a slow clock requires less imposed delay in order to put the system in unsynchronized mode.

It is should be mentioned that, in order to lead to sustainable consequences there is a minimum duration for this condition holding. This duration depends on the concrete application and the adversary goal. If the adversary imposes a delay only once, then during the next correction point the clock will be returned to the synchronized state. Therefore, the adversary needs to impose a selective delay as long as long he wants to keep nodes unsynchronized. Fig. 9.2 illustrates how an adversary can shift t_{clock} out of the allowed bounds and keep it there by imposing the same delay to each Delay Response message. The orange arrow shows the shift after the attack, whereas the green and red solid lines represent clock time within and out of the allowed bounds respectively. How far the clock time deviates from the bounds is shown by t_{out}.

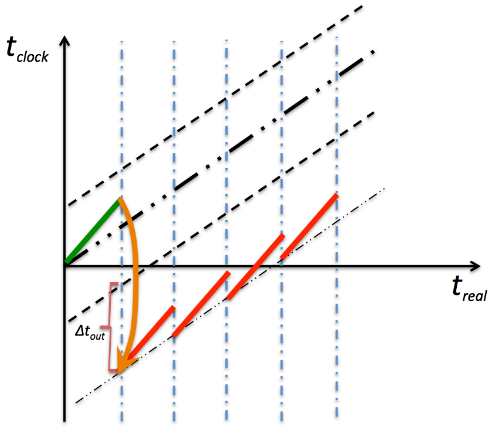

Fig. 9.2 Sustainable delay attack on the faster clock.

As an obvious countermeasure against a random delay attack, more frequent clock correction procedures can be used, but in case of a prolonged attack, different prevention and attack detection approaches are needed.

This analysis shows that one of the main assets for industrial networks can be breached quite easily, and become one of the main system vulnerabilities. This kind of attack can be performed under the assumption that the channel between two nodes can be controlled by an adversary. As it will be shown below, this assumption can be achieved by performing an ARP poisoning attack.

9.5. ARP Poisoning

ARP is a well-known and widely used protocol, and therefore, ARP poisoning is also a commonly used attack method [12]. The ARP protocol is part of the TCP/IP stack, and hence, it is used in many networks. To resolve the correspondence between MAC and IP addresses, ARP uses two types of messages: ARP request, a broadcast message, and ARP reply, a unicast message. The algorithm is simple: when a node A wants to send a message to a node B, and A has the IP address of B, it needs to ask about the corresponding MAC address. First A checks its ARP cash table, and if the address is not there, it sends a broadcast ARP request message to all network participants asking about the IP address it has. The node with the mentioned IP address answers with an ARP reply message, after which communication can start. These messages do not have any authentication properties, so it is easy to intercept and forge them. An adversary can send fake ARP replies to A saying that he is B and to B saying that he is A. After this A and B will communicate with each other through the adversary.

The ARP poisoning attack can be performed independently of the physical layer implementation, and therefore it is suitable for both wired and wireless networks. For instance, a possible scenario of applying ARP poisoning against an industrial network build on PROFINET IO was considered in [13]. In the paper, the authors demonstrate that by using an ARP poisoning attack, an adversary could gain control over the outputs of a PROFINET IO device, which can lead to a number of possible attack continuations with a huge impact on the network. In [14] the authors consider performing a combination of a hole 196 attack (an attack that uses a vulnerability of WPA2 and exposes the network for insider attacks) along with an ARP poisoning attack in networks based on the IEEE 801.11i standard. Such an approach allows the adversary to decrypt all traffic going through the access point from an attacked user. In both works mentioned above, the authors consider only the possibility of performing an ARP poisoning attack in different networks, whereas we, in our research, consider the specific use of an ARP poisoning attack to break synchronization.

9.6. The Attack Analysis

As it is possible to perform a man-in-the-middle attack in the network via ARP poisoning, it opens possibilities for the adversary to influence the clock synchronization algorithm by imposing a delay in the delay-request-response mechanism used by IEEE 1588. Such kind of an attack can lead to network synchronization failure. Depending on the link and node chosen for the attack, the consequences can range from one node failure to complete system disruption. The most obvious way to influence IEEE 1588 is to try to break one of its basic assumptions listed in Section 9.3.A. The second assumption is difficult to violate for an intruder as it mostly depends on system specific configurations and to be effective, the propagation delays should be significant. In contrast, the first assumption is appealing from an intruder point of view, as its violation implies an additional delay in only one direction and it does not need to be huge. Moreover, the message does not need to be changed — it should be only delayed. An ARP poisoning attack performed in a network using IEEE 1588 implies that the adversary has full control over the communication between two chosen nodes, even if the optional security extensions from Annex K are applied. This means that the adversary can easily impose the necessary delay in one of directions of the delay-request-response mechanism. The idea is that the adversary needs to influence the system in such a way that the output of the synchronization algorithm (the correction shift) is bigger than the allowed threshold.

A. Applicability

The described attack is possible even for networks using the optional Annex K of IEEE 1588. The security extensions provided in the annex include message integrity, group authentication and replay attack – however, message integrity cannot help against this attack, as the message does not need to be changed. Group authentication cannot help as the adversary is pretending to be a device already existing in the network. ARP poisoning implies a forge in the initial unprotected part of the communication between nodes, and later the messages between them would be passed normally without any change, apart from an additional delay caused in one direction. Replay attack protection also cannot help as the message is not supposed to be replayed.

There are some limitations for this type of attack and cases when the attack is complicated or useless. To perform this kind of attack successfully, an adversary needs to know which link, node/switch or set of nodes/switches to attack. This means that the adversary first needs to perform a network analysis to find the desired or the weakest point for attacking as attacking a random node most probably will not lead to costly consequences. Also, ARP attacks work only for subnetworks, and therefore, if the whole network consists of several subnetworks, the adversary cannot affect the entire network, as it cannot influence one subnetwork while performing the attack in another one. Another limitation is the network configuration. If we consider a network with a completely static configuration, then all network participants can get complete ARP tables with all addresses during the first initialization phase. This kind of attack is therefore possible only in bigger networks or networks where devices are allowed to join it during its operational phase.

On the other hand, the attack also has a number of advantages from an adversary point of view, where the two most appealing ones are that industrial networks of today do not consider this possibility and that the same technique can be used in several different types of industrial networks.

B. Consequences

As we consider industrial networks with different levels of criticality and complexity, some factors of the attack can depend on the specific use case and adversary goals. The scenario described above will put the two communicating nodes in an unsynchronized state. This is a straight forward case, but if the industrial system is developed to consider possible faulty nodes for increased robustness, which is often the case in systems with high cost of failure, the adversary has to target the disruption of a set of nodes. The size of this set depends on the application and the network architecture. Thus it is rational to target a node that is critical to the system functionality.

Within the considered network, an adversary can target a grandmaster clock or a slave clock. If the grandmaster clock is put into an unsynchronized state, the system will choose a new grandmaster clock according to the BMC algorithm. In this scenario, the adversary can influence the overall network performance, e.g., if the network has only a limited amount of clocks with external GPS receivers, the adversary can aim to remove these from the network first, by putting them into unsynchronized mode. This way, the network will degrade considering clock synchronization precision. It is worth to mention that in order to keep a clock in unsynchronized mode, the adversary needs to keep influencing the propagation delay, or else only a transient clock synchronization error occurs.

If the adversary targets a slave clock, putting it in unsynchronized mode will not influence the others clocks. This can therefore be beneficial for the adversary only in case the corresponding node has a critical functionality and the system does not have any redundancy. Note that even after detection of an unsynchronized node, the reason for becoming unsynchronized will not be discovered unless the system has countermeasures against ARP poisoning. Therefore, even if maintenance functionalities will replace the node thinking that it is out of order, the adversary can simply continue to influence it in a similar way. In case a node is critical, it is also interesting to investigate if and when its unsynchronized behavior would be detected by the system.

The adversary can be interested in transient system influences, i.e., keeping a clock unsynchronized for a short period of time, if he needs to masquerade or hide some other short time activity that otherwise can be detected. For example, if the adversary wants to sabotage an assembling line on a plant, he can target the pressure or distance sensor nodes. Their unavailability, even for a short period of time, can lead to an accident. This scenario is important for critical applications where availability is one of the main security objectives.

The cases described above demonstrate that the ARP poisoning attack is problematic for industrial networks that do not have any kind of protection against it. Further, it shows that an adversary can pursuit several different goals by conducting the same attack. This fact makes the considered combination of attack techniques even more appealing for an adversary.

The cases described above demonstrate that the ARP poisoning attack is problematic for industrial networks that do not have any kind of protection against it. Further, it shows that an adversary can pursuit several different goals by conducting the same attack. This fact makes the considered combination of attack techniques even more appealing for an adversary.

9.7. Overview of Mitigation Techniques

In this section, we give an overview of existing solutions and evaluate them from an industrial point of view. There are plenty of different security solutions, covering different sets of security objectives, applied at different layers of OSI stack, suitable for different environments etc. [2]. We consider the possibility to break clock synchronization by performing an ARP poisoning attack and thereafter imposing delays on a hijacked communication channel. Due to this, the security solutions can be divided into the two main categories: mitigation techniques against ARP poisoning and mitigation techniques against delay imposing. Solutions from both categories can be used for clock synchronization protection.

A. Mitigation techniques against ARP poisoning

ARP poisoning is a well-known type of attack, and therefore, possible countermeasures were investigated in many research papers. We consider the most relevant ones for this particular use-case and investigate if and how they can be applied in industrial applications.

A comparative analysis of possible mitigation techniques for ARP poisoning was presented in [6], and in [5] [15] several protection techniques were presented. Most existing techniques like [5, 6] and [15] imply implementation of an additional security mechanism. It can be e.g., an added encryption scheme, i.e., the node can encrypt all ARP request and reply messages. This solution helps against the ARP poisoning attack, but it also means additional computational power in the nodes, and additional delay for messages transmission. The additional overhead is disadvantageous for industrial applications with low latency requirements, as it requires complementing all network participants with an encryption/decryption module which lacks backwards compatibility.

The second approach is the introduction of a control element in the network that monitors and analyses it in order to prevent a possible ARP poisoning attack. This can be realized via a centralized detection and validation server [5, 16] or a passive analyzing detection system. The server can confirm ARP request and validate the ARP tables of all the nodes within the network. The applicability of this method in industrial environments depends on the network size, since the process of controlling all ARP tables in a huge network can become problematic, and also implies the introduction of a single point failure, since if the server is compromised or fails, the protection stops working. This is a questionable solution for critical applications and for applications with a distributed control architecture. The passive detection system looks promising, but it also has its limitations. Such a detection system can record all ARP requests and replies and construct the network according to the information gathered. It can constantly monitor the resulting network map for inconsistency that will indicate an attack. This approach can work only if the attack starts after the data analysis has been initiated, and thus this method is also limited by the size of the network. However, the approach can be a good candidate for mixed protection systems, where we combine several methods in order to achieve an appropriate overall security level.

Another approach implies using an Intrusion Detection System (IDS) with probe messages [15]. Classical IDS detects an intruder by monitoring the system states, and thus this approach works under the assumption that the intruder causes a difference in the state sequences that can be detected by the IDS. Therefore, a classical IDS cannot be used against ARP poisoning, as the attack does not cause any difference in the event sequence. Hence, to be used against ARP poisoning, IDS should be complemented with a probe message mechanism. The idea is that the control-monitoring center from a classical IDS can now send probe messages to the network participants and these messages cause a difference in event state depending on presence or absence of a malicious adversary performing ARP poisoning. Requests from the monitoring center to verify genuineness of ARP request and replies can be used as such probe messages. This approach introduces additional communication overhead, implying that for delay-sensitive systems and large networks, using IDS can be expensive. In addition, the method leads to the introduction of a single failure point.

B. Mitigation techniques against delays imposing

The main idea of protection against delay imposing is to check and control propagation delay [10]. The method implies monitoring performed by the node itself or a switch. In this case, the network participant needs to collect and analyze statistics about message propagation delays in the networks. However, the approach requires additional recourses that can be complicated if the participant is a wireless sensor node. Further, the approach works only if the adversary joins the network after the statistic collection and analysis has begun. This mitigation technique has a probabilistic character.

Considering the known techniques against ARP poisoning and delay imposing, we can see that they do not fully suit the industrial environment. Therefore, a possible solution in this case can be a combination of several approaches. A combined approach can be considered as a defense-in-depth technique, because initiation of the second category assumes failing of the first one. This can bring flexibility and allow satisfying the requirements needed for a possible security solution. For example, for most small networks, we can use encryption, while in less critical parts of the network, an intrusion detection system along with a delays analysis can be applied.

9.8. Evaluation of Impact with ARP Poisoning

In this section we present the results of our evaluation of the impact of an ARP poisoning attack targeting the clock synchronization functionality. The evaluation process can be separated into two steps. The first step concerns the ARP poisoning itself, by formally specifying the ARP protocol and possible adversary actions. The second step is to evaluate the breaking of clock synchronization in the system assuming that the ARP attack was performed.

The tool used for the evaluation in this paper is Automated Validation of Internet Security Protocols and Applications (AVISPA) [17]. AVISPA is typically used for sensitive security protocols and application evaluation and analysis. It uses High-Level Protocol Specification Language (HLPSL) for interactions with users. The Security Protocol Animator (SPAN) [18] tool was developed to simplify the interaction process with a user. SPAN allows a user to use CAS+ language to describe the protocol and then it translates it to HLPSL. This makes the work with the AVISPA easier, since all a user needs in order to analyze a protocol is to specify the modules: identifiers, messages, knowledge and goals. To formalize our proposed attack process, the following situation was considered:

Identifiers. We use the simplest case, when there are three participants in the network A, B, and C. A and B are benign network participants and C is an adversary. As it is shown in Table 9.1 they are specified as users, whereas IP and MAC addresses of all three are specified as numbers.

Table 9.1 Identifier Declaration

Type	**Identifier**
User	A, B, C
Number	IPa, IPb, MACa, MACb, MACc

Messages. The specified message set implies that the adversary sends ARP responses to both A and B with the wrong IP addresses, i.e., C sends an ARP response with the IP address of B and MAC address of C to A and correspondingly the IP address of A and MAC address of C to B.

Knowledge. Each network participant knows all IP addresses in the network plus its own MAC address (Table 9.2).

Table 9.2 Identifier Declaration

User	**Knowledge**
A	C, B, IPa, MACa
B	A, C, IPb, MACb
C	A, B, IPa, MACc, IPb

Goal. The tool is limited in the definition of possible goals, so we apply the reverse technique to its formulation, i.e., in case the tool proves that the goal is achieved, it means that clock synchronization is broken. To prove that ARP poisoning can be performed, we specify the goal as: to keep the secrecy of the MAC address of B from A. If the tool shows that the protocol is safe, this means the adversary wins, as A cannot understand that he is communicating with C instead of B.

The assumption we use in the modeling is that the intruder (node C) knows the IP addresses of the targeted network participants. If the network allows a new device to join, then node C can be considered as a new device in the network, and otherwise we assume that node C was in the network already from the initialization phase.

AVISPA has several types of analysis techniques, namely the On-the-fly Model Checker (OFMC), the Constraint-Logic-based Attack Searcher (CL-AtSe), the SAT-based Model Checker (SATMC) and the Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). We used OFMC in our evaluation, as this technique can prove that the specified protocol is correct, and this is exactly what we need, given the way we defined the goal and that we included an intruder in the users list (node C). OFMC [19] explores the transition system by using a demand-driven approach. The checker uses symbolic techniques, a lazy Dolev-Yao intruder model and lazy data types. The last one means that data constructors do not evaluate data arguments while building it, which allows infinite data computing. This attacker model is suitable for our approach, as in our modeling we assume that an intruder is a legitimate network participant.

The analysis with OFMC shows that an ARP poisoning attack is possible under the given assumption, namely that the adversary knows the IP addresses of network participants, implying that some prior network analysis has been conducted.

The second step of the evaluation can be done using logical reasoning without any verification tool. The clock synchronization algorithm can be broken if an adversary succeeds in breaking one of its basic assumptions. In our case, the assumption is that the propagation delay is equivalent in both directions within the same logical channel. Obviously, if the adversary successfully performed a man-in-the–middle attack and controls the communication process in both directions, he can impose the necessary delay in one direction. More precisely, the adversary needs to impose a delay greater than the maximum allowed clock drift. This can be done if the adversary knows the synchronization period and the maximum allowed clock drifts in the system. This knowledge, as well, can be gained through prior network and specific application analysis.

9.9. Conclusions and Future Work

In the paper we investigated the possibility to break the network clock synchronization mechanism established according to IEEE 1588 standard by performing an ARP poisoning attack. We also considered possible mitigation techniques that can protect the network from the attack or can protect the clock synchronization even in case of a successfully performed ARP attack. The mitigation techniques take into account the derived requirements from the targeted industrial applications. The evaluation using AVISPA showes that this scenario is indeed possible. The result indicates the need to develop a suitable security solution that can be incorporated in a security framework and that should satisfy the requirements derived from industrial applications.

As future work, we plan to continue this investigation by looking at possible solutions and develop a technique that can protect clock synchronization in industrial applications.

Acknowledgments

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n°607727. E. Uhlemann is partly funded by the Knowledge Foundation through the WIRE project. E. Uhlemann is partly funded by the Knowledge Foundation through the WIRE project.

Bibliography

[1] H. Kopetz and W. Ochsenreiter, "Clock Synchronization in Distributed Real-Time Systems," *IEEE Transactions on Computers,* vol. C-36, no. 8, pp. 933-940, 1987.

[2] E. Lisova, E. Uhlemann, J. Åkerberg, and M. Björkman, "Towards secure wireless TTEthernet for industrial process automation applications," *in Proc. ETFA*, Barcelona, Spain, Sep., 2014.

[3] IEEE. (2008). *IEEE 1588, "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems"*. Available: http://www.nist.gov/el/isd/ieee/ieee1588.cfm

[4] A. Treytl, G. Gaderer, B. Hirschler, and R. Cohen, "Traps and pitfalls in secure clock synchronization," *in Proc. IEEE ISPCS*, Vienna, Austria, Oct. , 2007.

[5] S. Kumar and S. Tapaswi, "A centralized detection and prevention technique against ARP poisoning," *in Proc. CyberSec*, Kuala Lumpur, Malaysia, Jun, 2012.

[6] N. Tripthi and BM Mehtre, "Analysis of various ARP poisoning mitigation techniques: a comparison," *in Proc. ICCICCT*, Kanyakumari, India, Jul., 2014.

[7] C. L. Abad and R. I. Bonilla, "An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks," *in Proc. ICDCS*, Toronto, Ont., 22-29 Jun., 2007.

[8] D. Dzung, M. Naedele, T. P.Von Hoff, and M. Crevatin, "Security for Industrial Communication Systems," *Proc. IEEE,* vol. 93, no. 6, pp. 1152-1177, 2005.

[9] J.-C. Tournier and O. Goerlitz, "Strategies to secure the IEEE 1588 protocol in digital substation automation," *in Proc. CRIS*, Linkoping, Sweden, Mar., 2009.

[10] M. Ullmann and M. Vögeler, "Delay attacks - implication on NTP and PTP time synchronization," *in Proc. ISPCS*, Brescia, Italy, Oct., 2009.

[11] C. Önal and H. Kirrmann, "Security improvements for IEEE 1588 Annex K: Implementation and comparison of authentication codes," *in Proc. ISPCS* San Francisco, CA, Sep., 2012.

[12] D. Bruschi, A. Ornaghi, and E. Rosti, "S-ARP: a secure address resolution protocol," *in Proc. ACSAC*, Las Vegas, NV, Dec., 2003.

[13] J. Åkerberg and M. Björkman, "Exploring security in PROFINET IO," *in Proc. COMPSAC*, Seattle, WA, Jul., 2009.

[14] F. T. Sheldon, J. M. Weber, S.-M. Yoo, and W. D. Pan, "The Insecurity of Wireless Networks," *Security & Privacy,* vol. 10, no. 4, pp. 54-61, May 2012.

[15] F. A. Barbhuiya, S. Biswas, and S. Nandi, "An active DES based IDS for ARP spoofing," *in Proc. IEEE SMC*, Anchorage, AK, Oct., 2011.

[16] A. P. Ortega, X. E. Marcos, L. D. Chiang, and C. L. Abad, "Preventing ARP cache poisoning attacks: A proof of concept using OpenWrt," *in Proc. LANOMS* Punta de Este, 19-21 Oct., 2009, pp. 1-9.

[17] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. V. Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Vigano, and L. Vigneron, "The AVISPA tool for the automated validation of internet security protocols and applications," *in Proc. 17th CAV*, Edinburgh, Scotland, Jul., 2005.

[18] Y. Glouche, T. Genet, O. Heen, and O. Courtay, "A security protocol animator tool for AVISPA," *in Proc. ARTIST2*, Pisa, Italy, May, 2006.

[19] D. Basin, S. Mödershein, and L. Vigano, "OFMC: A symbolic model checker for security protocols," *International Journal of Information Security,* vol. 4, no. 3, pp. 181-208, Jun. 2005.

Chapter 10

Paper D: Protecting Clock Synchronization – Adversary Detection through Network Monitoring

Elena Lisova, Marina Gutiérrez, Wilfried Steiner, Elisabeth Uhlemann, Johan Åkerberg, Radu Dobrin, and Mats Björkman.

In Journal of Electrical and Computer Engineering, special issue "System and Network Security: Anomaly Detection and Monitoring", Hindawi, April 2016.

Abstract

Today, industrial networks are often used for safety-critical applications with real-time requirements. The architecture of such applications usually has a time-triggered nature that has message scheduling as a core property. Real-time scheduling can be applied only in networks where nodes share the same notion of time, i.e., they are synchronized. Therefore, clock synchronization is one of the fundamental assets of industrial networks with real-time requirements. However, standards for clock synchronization, i.e., IEEE 1588, do not provide the required level of security. This raises the question about clock synchronization protection.

In this paper we identify a way to break synchronization based on the IEEE 1588 standard by conducting a man-in-the-middle (MIM) attack followed by a delay attack. MIM attack can be accomplished through e.g., Address Resolution Protocol (ARP) poisoning. Using AVISPA tool we evaluate the potential to perform an ARP poisoning attack. Next, an analysis of the consequences of introducing delays is made, showing both that the attack can, indeed, break clock synchronization and that some design choices, such as a relaxed synchronization condition mode, delay bounding and using knowledge of environmental conditions, can be made to make the network more robust/resilient against these kinds of attacks.

Lastly, network monitoring is proposed as a technique to detect anomalies introduced by an adversary performing attacks targeting clock synchronization. The monitoring capabilities are added to the network using a *Configuration Agent*, which, based on data obtained from the network, is able to detect an attack. The main contribution of the paper is a detailed problem description and evaluation of a security vulnerability in IEEE 1588 against delay attacks together with an evaluation of several approaches as possible mitigation techniques for the attack.

10.1. Introduction

One of the specific characteristics of industrial networks is a high cost of failure, resulting in money loss, environmental threats or damage to humans. Today, such networks grow extremely fast in complexity and functionality leading to an increasing number of security requirements [1]. Developing a security framework for the whole network implies taking into account application specific features and related systems assets [2]. Clock synchronization is an essential part of all networks with real-time requirements. Basically all industrial networks have real-time requirements, and therefore most messages have deadlines to meet. Consequently, if there is a way to breach clock synchronization, it will disrupt the network functionality, and moreover, it can be applied to a range of different networks regardless of their specific application area [3]. This fact leads to an increased possibility for an adversary to invest resources in such attacks. Also, it is a motivation to investigate possible ways of clock synchronization protection, which includes adversary detection and consequences mitigation.

Most synchronization algorithms, e.g., the IEEE 1588 standard, are vulnerable to delay attacks, as they rely on measuring delays without taking adversary attacks into consideration. A possible way to breach clock synchronization was proposed in [4], namely a combination of an ARP poisoning attack followed by a delay attack. To conduct a delay attack, an adversary first needs to penetrate the network. A so called man-in-the-middle (MIM) attack is one possible way to take control over a communication channel. When performing a MIM attack, the adversary is positioned inside the communication channel between benign participants, e.g., through ARP poisoning. An ARP poisoning attack takes advantage of the vulnerability in the ARP protocol, which is used for translating between IP and MAC addresses. If an adversary can convince a benign network participant N_1 to connect the adversary MAC address to the IP address of another benign node N_2, with whom N_1 wants to communicate, the adversary will get all traffic sent by N_1 to N_2. When the adversary controls the channel, the next step in order to break clock synchronization is to perform a selective delay attack. The combination of these two techniques can be used to breach clock synchronization.

An intrusion detection system (IDS) entails monitoring a network for security purposes. This is not a new approach, and a Network Security Monitor (NSM) system was described already in 1994 [5]. Statistics gathered from a network can be used for many functionalities, such as QoS enhancement, network adaptation for having higher resilience levels, routing optimization, etc. Usually, an adversary penetrating the network changes its behavior in some way. While a passive adversary mainly monitors and analyzes incoming data, the gathered statistics can be used by an active adversary in an upcoming intelligent attack. In case of an active adversary, communication with other network participants is also common. This means that the adversary will introduce new traffic and/or change the traffic pattern. Therefore, if the Monitor, collecting and analyzing the network statistics, can detect a deviation, the adversary presence can be discovered. It is the first step in network protection, which should be followed by adversary isolation or neutralization and consequences mitigation.

The main contribution of the paper is a detailed problem formulation of clock synchronization vulnerabilities along with evaluation of possible techniques to mitigate consequences of clock synchronization breaking. The investigation of the possibility to detect a malicious adversary targeting clock synchronization breaching in industrial networks using IEEE 1588 for clock synchronization was also conducted. The range of possible attacks is narrowed down to a delay attack performed after a successfully conducted MIM attack. Two scenarios for conducting a MIM attack via ARP poisoning with single and multiple network penetrations are considered. Such choice of scenarios allows to conduct a discussion about the difference in detection and the following adversary localization once they are formally evaluated in AVISPA [6]. ARP poisoning attack is a well known one, but we decided to conduct its formal evaluation, as such approach allows evaluate possible mitigation techniques in future in the same way and compare possible solutions. Further, two different ways of performing a delay attack are simulated in OMNeT++ [7] in order to see how these ways can affect traffic characteristics. The results are also discussed from two points of view: the adversary and the Monitor. Our results show that the likelihood of adversary detection depends on many factors, such as the prior knowledge of the network available to the adversary, the knowledge of the network history available to the Monitor, and, the ability of the network to switch to Relaxed Mode, i.e., allowing additional clock drifts. Environmental conditions are also considered as an additional factor for clock synchronization disturbance and a technique using these conditions for adversary detection is proposed. Finally, another technique that can detect the presence of an adversary, namely delay bounding, is discussed and evaluated

The remainder of the paper is organized as follows. Sections 10.2 presents related works and Section 10.3 introduces some background information regarding clock synchronization, in particular the IEEE 1588 standard, and IDS. Next, the system model including models of the adversary and the Monitor are described in Section 10.4. In Section 10.5, the MIM attack is investigated along with the possible consequences of a delay attack for clock synchronization, whereas Section 10.6 describes the proposed solution for attack detection and countermeasures discussion. Results of the attack evaluation and monitoring simulations are presented in Section 10.7. Finally Section 10.8 concludes the paper.

10.2. Related Works

The initial version of the IEEE 1588 standard does not have any security services. Based on this the authors of [8] show the effects of a delay attack on the IEEE 1588 Precision Time Protocol (PTP). Our work can be considered as an extension of mentioned above paper, as we propose the way of conducting delay attack and widen possible mitigation techniques. In release 2008, Annex K was added to the standard to provide a set of security solutions [9]. However, the amendment provides only a very limited set of services. Annex K provides guidelines for message integrity protection and group source authentication. These security solutions do not help against a delay attack, as in this case the adversary does not need to change the messages or create new ones - it simply delays them. Mizrahi applied game theoretical approach to analyze the delay attack influence on clock synchronization [10]. As a result, the author proposed to use a multiple-path approach to prevent and mitigate delay attacks. However, this approach is not compatible with IEEE 1588.

In [11], network monitoring has been used in the context of synchronized networks. The concept of a Configuration Agent is introduced: an autonomous entity that learns the characteristics of the network through continuous monitoring, with the goal of facilitating the configuration and re-configuration of time-triggered networks. The Configuration Agent is composed of four elements: Monitor, Extractor, Scheduler and Reconfigurator. The duple formed by the Monitor and the Extractor gathers data from the network and distills relevant information from it. In [12], that information is sent to the Scheduler so it can produce a new schedule for the network, with which the network is re-configured. For this paper, we propose to use a generalized view of the same concept, so we replace the Scheduler with a Diagnoser, which after the learning phase performed by the Monitor and the Extractor, will use the distilled information to decide what actions to take. Finally the Reconfigurator carries out those actions. An example of a network security Monitor running on Ethernet and applied for LANs is presented in [13]. The authors proposed an hierarchical model of data analysis that allows separation of network activities into host-to-host, services, and connections groups. In [14], traffic patterns are proposed to be used for detection of periodic communication of Botnets, subnetworks consisting of infected devices remotely observed by the adversary. These two examples are from different areas and are separated significantly in time, demonstrating that security and monitoring can complement each other in an efficient way.

As mentioned above, applying network monitoring techniques to security issues leads to the development of an IDS. There are two main types of IDS depending on the logic of detection [15]. The first one is a signature-based network IDS. In this approach, there is a set of known attacks together with their corresponding patterns. The IDS is monitoring the system and raises an alarm, when there is a system pattern matching the one from the set. This approach has an obvious limitation, if there is an attack that was not considered at the development phase, it will not be detected. The second group is called heuristic or anomaly-based IDS. With this approach, the system instead knows some standard ways of behavior of the network and search for any anomaly, anything that does not match the standard pattern. The advantage of this method is the possibility to detect a previously unknown attack. On the other hand, if the intruder knows the specific network patterns, the adversary actions can be masked and indistinguishable from the normal network behavior. The patterns of communication in conjunction with clock synchronization algorithms, which are in the main scope of this paper, are well known. Therefore, in our case we are targeting a hybrid technique, i.e., combining a heuristic and a signature-based method. In this case, the Monitor can be more flexible and have a higher probability of an attack detection.

When evaluating a new solution or mitigation technique, it is essential to have some benchmarks and evaluation criteria. In [16], the authors propose a metric-based approach for IDS evaluation. Logical, architectural and performance metrics were presented as the main groups of criteria. Logical metrics imply such characteristics as cost, maintainability and manageability. Adjustable sensitivity, data pool scalability, data storage, and similar, can be considered as architectural metrics. Finally, error reporting and recovery, induced traffic latency, operational performance impact, and observed false positive and negative ratios, represent performance metrics. The scope of Monitor evaluation used in this paper is to show the impact of the Monitor on the network and the efficiency in attack detection.

10.3. Background

A. Clock Synchronization

In order to be able to cooperate, nodes of industrial networks have to share the notion of time, i.e., be synchronized. In the ideal case, every node has perfect clock and simply follows the schedule based on its time. In reality, each clock has a natural drift. This drift can be different, mostly depending on the cost of the clock: usually the more expensive the clock is, the more accurate it is. Clock synchronization algorithms are used to assist the nodes with clock correction. Clock drift is a natural characteristic caused by the underlying physical oscillators Therefore, it cannot be completely eliminated, only mitigated, i.e., periodically compensated for. In industrial applications, often only the relative clock differences are important for network correct performance. Consequently, such clocks should be synchronized to each other rather than to an external time reference (e.g., such as GPS).

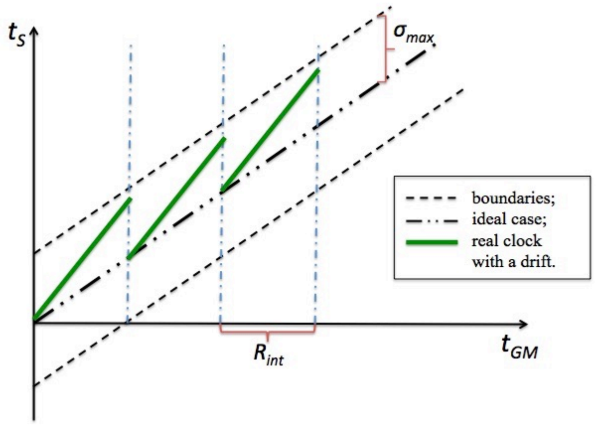

Fig. 10.1 Periodic correction of slave clock time, t_S according to grandmaster clock time, t_{GM}.

Clock correction is done periodically as is shown in Fig. 10.1. The purpose of clock correction is to keep clock time within acceptable boundaries (dashed black lines). The ideal case would be that times provided by the grandmaster clock and the slave clock are always the same (black line with two dots and a dash). But in reality, the slave clock will drift apart from the grandmaster clock (solid green line) and therefore, it needs to be periodically corrected.

B. IEEE 1588 standard

Given two clocks, *A* and *B* in a network such as they have been synchronized in a moment in the past, i.e., $t_A = t_B$, in a later moment the time values provided by these clocks will have drifted apart such as:

$$\left|t_A - t_B\right| = \sigma \ . \tag{10.1}$$

This drift is caused by the non-ideality of the clocks and environmental conditions, e.g., heat affecting the frequency of the oscillators.

To avoid failures, caused by inability of applications to meet their deadlines, clock synchronization protocols are used. Among those, IEEE 1588 is a standard widely used in industrial applications for providing and maintaining clock synchronization [17]. In the IEEE 1588 standard, one of the nodes is chosen as a grandmaster *GM*, and the rest of the nodes are referred as slaves S_i. Slaves are synchronized to the grandmaster, such that the differences in time values provided by the local clocks in the nodes, as expressed in (10.1) is bounded. This is expressed in the synchronization condition:

$$\left| t_{S_i} - t_{GM} \right| < \sigma_{\max} \qquad (10.2)$$

that should be constantly preserved. Here, σ_{max} is a parameter that should be chosen so that the application requirements are satisfied. The minimum value of σ_{max} can be calculated as:

$$\min(\sigma_{\max}) = 2\max(\rho_i)R_{\text{int}} \qquad (10.3)$$

where ρ_i are the clock drifts of the clocks in the network and R_{int} is the re-synchronization interval. In Fig. 10.1, it can be seen how IEEE 1588 works by periodically adjusting the value of the slave clock. The period of these corrections is the resynchronization interval. The value of the offset used to correct the slave clock is calculated applying the protocol depicted in Fig. 10.2 The clock synchronization protocol consists of a series of messages being exchanged and time-stamped between the grandmaster and the slave in order to gain enough information to calculate the offset. This process is repeated periodically every re-synchronization interval R_{int}. The message exchange is as follows:

1) At $t = t_1$ the grandmaster sends a synchronization message, (`sync` in Fig. 10.2) containing t_1 to the slave.

2) At $t = t_2$ the slave receives the `sync` message. Now both t_1 and t_2 are recorded in the slave.

3) At $t = t_3$ the slave sends out the delay request message (`delay_req` in Fig. 10.2). t_3 is also recorded in the slave.

4) At $t = t_4$ the grandmaster receives the `delay_req` message.

5) At $t > t_4$ the grandmaster sends the delay response message (`delay_resp` in Fig. 10.2) containing t_4 to the slave. When the slave receives the `delay_resp` message, lastly, t_4 is recorded.

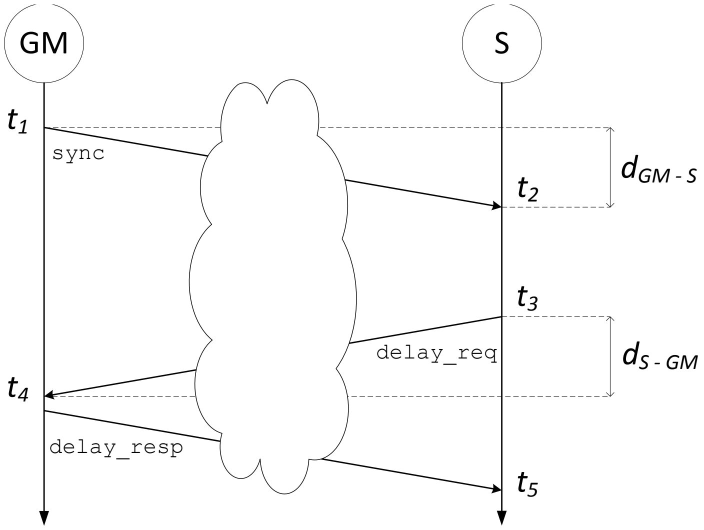

Fig. 10.2 Clock synchronization protocol.

Finally when all the time-stamps have been collected by the slave, the offset, σ_{meas}, can be calculated according to

$$\begin{aligned} d_{GM\to S} + \sigma_{meas} &= t_2 - t_1, \\ d_{S\to GM} - \sigma_{meas} &= t_4 - t_3, \end{aligned} \tag{10.4}$$

where $d_{GM\to S}$ and $d_{S\to GM}$ are the transmission delays of a message going from the grandmaster to the slave and from the slave to the grandmaster respectively. Now if we assume that the transmission delay is symmetric, i.e., it is the same in both directions, then $d_{GM\to S} = d_{S\to GM} = d_0$ and the measured offset is:

$$\sigma_{meas} = \frac{1}{2}((t_2 - t_1) - (t_4 - t_3)) \ . \tag{10.5}$$

Ideally, this value of the measured offset reflects the difference between the grandmaster clock and the slave at the moment when the offset is measured

$$t_S - t_{GM} = \sigma_{meas} \ . \tag{10.6}$$

In the IEEE 1588 standard, the equation for offset calculation is more complicated, as there are parameters compensating the propagation delays. For the sake of simplicity they are omitted here, as they do not change the logic of the protocol and do not make any significant difference for conducting a delay attack.

The standard defines three types of clocks, they are *transparent*, *boundary*, and *ordinary*, respectively. A transparent clock performs hardware time-stamping of synchronization messages and updates the corresponding fields in them. A boundary clock has one of its ports in slave mode and gets the time from the grandmaster via this port. It does not update synchronization messages, but can create new ones with the time-stamps according to the information provided by the slave port and send it out. An ordinary clock is a clock without any specific additional functions.

In addition, the standard defines two possible operation modes and two modes for synchronization messages exchange. The following operational modes are possible: end-to-end and peer-to-peer. In the first mode, clocks get information about the delays in links from the exchange of synchronization messages each time they want to make a correction. In the second mode, this exchange of messages is performed for all links regularly and without relation to the clock correction events. Each time a clock wants to correct its time, it has information about all delays with all neighbors. The end-to-end operational mode is suitable for networks where it cannot be guaranteed that all devices in the network support IEEE 1588. The peer-to-peer mode implies that the IEEE 1588 standard is supported by all devices in the network. All above mentioned synchronization message exchanges can be performed in two or in four steps. In the second variant, follow-up messages are used to provide more precise time-stamps.

10.4. System Model

Considering possible ways of attacking the system and analyzing system reaction to the intrusion, it is important to set the limits and assumptions of investigating scenarios. There are many possible ways of interactions between the adversary and the system depending on the assumptions for both. In this paper we consider wired networks, where in synchronization is established and maintained according to IEEE 1588. Using peer-to-peer mode in the network implies that network participants periodically exchange messages to be aware of delays in the channels between them and their neighbors. The networks consist of routers capable of messages time-stamping and nodes, particularly grandmasters and slaves. Routers and nodes have the transparent type of clocks, this means that they perform hardware time-stamping of synchronization messages upon arriving and transmitting, via update of the correction field in the follow-up messages.

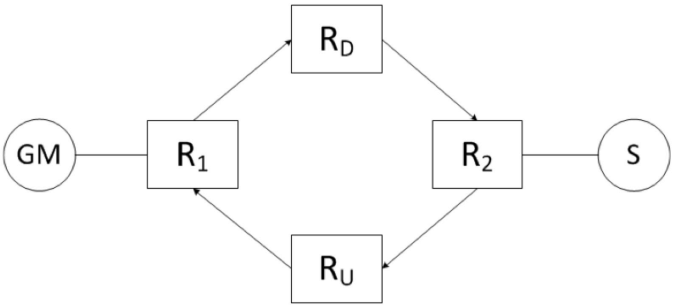

Fig. 10.3 The Network topology used for the simulations.

Fig. 10.3 shows the sample topology. The network consists of a grandmaster, a slave, and a set of routers. There are two communication channels between the grandmaster and the slave, namely downlink and uplink.

A. Adversary model and goal

To perform an attack analysis, the adversary model should be specified first. We assume that the adversary has access to and initial knowledge about the network. The adversary knows which node or communication link it is going to attack. The adversary targets clock synchronization breaking, therefore, it attacks a link which is involved in the synchronization protocol. Except the assumptions mentioned above, adversary choices can be random or based on an analysis of network conducted in advance. We consider a case when the adversary attacks only communication channels, links. In this case, the adversary is capable of receiving, transmitting and delaying messages. At this point, the capability of learning (i.e., possibility to analyze the reaction of the opponent and connect consequences with causes) and behavioral adaptation is not considered. The main adversary goal is system disruption, i.e., the adversary intends to cause system error and propagate it as much as possible, so that it leads to a system failure. Also, the adversary targets to prolong its influence and stay undetected. We assume that the behavior is rational in pursuing the abovementioned goals.

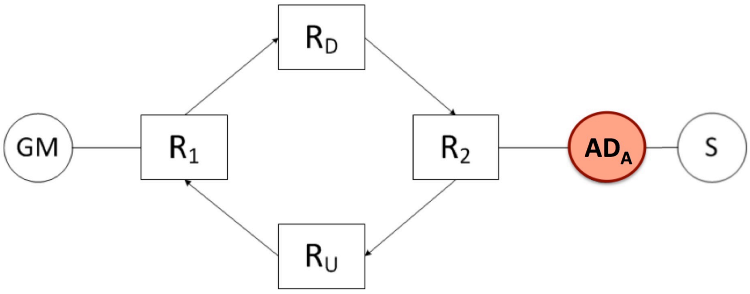

Fig. 10.4 Scenario *A* for the network, AD_A - an adversary.

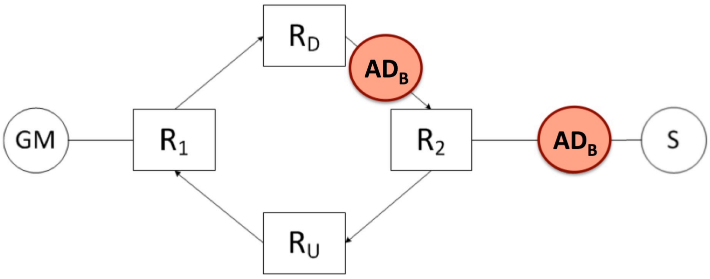

Fig. 10.5 Scenario *B* for the network, AD_B - an adversary.

We investigate the case when the adversary uses an ARP poisoning attack to penetrate the network and take control over communications in the targeted channel, i.e., the adversary conducts a man-in-the-middle attack. The next phase for the adversary is to perform a selective delay attack targeting synchronization. As it was specified in the adversary description, the objective of the attack is a link. In this paper we consider two scenarios as is shown in Fig. 10.4Fig. 10.5. Scenario *A* is a case when one link is under attack. In this case the adversary controls the communication between a router R_2 and a slave *S*. The second considered case, scenario *B*, is a consecutive attack on two links. This case is represented by AD_B in Fig. 10.5, the adversary controls communication channels between the router R_D and the router R_2 and between the router R_2 and the slave *S*. These two links under attack are the parts of one logical channel between the grandmaster *GM* and the slave *S*. This scenario implies that the adversary can interfere with the targeted message in different parts of its propagation path ($R_D \rightarrow R_2$ or $R_2 \rightarrow S$). This difference is important for the choice of mitigation techniques. Even though for the slave the results will look exactly the same, during delays analysis it can be more difficult to distinguish the adversary from natural network disturbances. Furthermore, after the adversary detection, it can be more difficult to locate the adversary. Also, such a scenario can be a basis for future modeling of a compromised router. It can be achieved if the possibility to change the contents of the messages is added to the adversary skills set, as then the adversary an actually replace the router from a functional point of view.

B. Configuration Agent Model

The four elements that compose the Configuration Agent that we propose to introduce in the network to detect possible attacks can be seen in Fig. 10.6. First, the Monitor gathers traffic measurements. Next, the Extractor transforms these traffic measurements in relevant traffic parameters. Furtherafter, by analyzing the traffic parameters, the Diagnoser is able to detect if there are anomalies in the traffic patterns and it determines which are the correct actions to take. Finally, the Reconfigurator changes the configuration of the network to introduce the changes proposed by the Diagnoser.

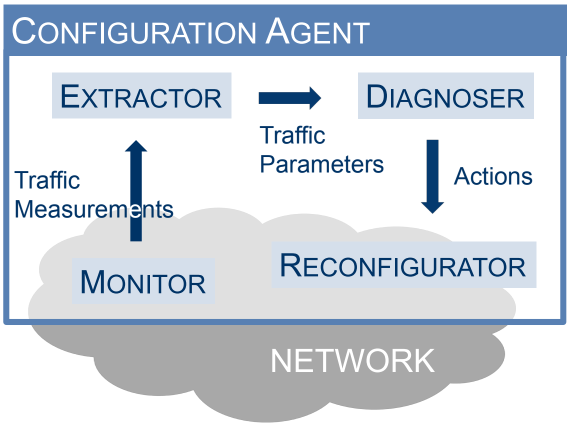

Fig. 10.6 Configuration Agent overview.

In this paper, we assume that all the functionalities of the Configuration Agent take place locally in every slave in the network. This means that the only information that the Configuration Agent has is the one gathered by the Monitor in a given slave. This approach presents both limitations and advantages. On the one hand it guarantees that the detection and mitigation process is not affected by the same attack that we are trying to detect. On the other hand, having a global view of the network, that allows us to combine the information gathered in every slave, can certainly help with the detection of the attack. That scenario will be explored in future works.

10.5. Vulnerability Analysis

A. ARP poisoning as a method of performing a MIM attack

ARP is a network layer protocol used to define the correlation between MAC and IP addresses for the network participants. When a node N_1 wants to send a message to a node N_2, N_1 knows the IP address of N_2, but to send the message it also needs to know the related MAC address. N_1 first checks its table of IP and associated MAC addresses. If it cannot find the MAC address of N_2 in the table, it sends out a broadcast message requesting the node that has the IP address of N_2 to reply and send back its related MAC address. This communication is completely unprotected and hence vulnerable. Whenever any node receives an ARP reply, it overwrites its table even if it has not sent the request. A MIM attack is an attack when an adversary controls the communication channel between two parties. The parties believe that they communicate with each other directly, but in reality all data exchange is going through the adversary, who possibly can influence the data. An ARP poisoning attack is an attack using the ARP algorithm vulnerability to perform a MIM attack. A malicious adversary can send an ARP response to N_1 pretending to be N_2, and one to N_2 pretending to be N_1. As a result, communication between N_1 and N_2 will go through the adversary. Even though this is a well known attack, it still remains valid and possible, and is used for network penetrations [18].

B. Possible targets of the attack

ARP poisoning re-directs traffic and allows the adversary to control communications between specific sets of MAC addresses. Depending on the adversary goal and the specific application, there are two possible targets of such an attack. Either the adversary can target a concrete communication link, hereafter referred to as scenario *A*, or it can target a specific device in the network, implying that more than one link needs to be compromised, termed scenario *B*. In case of scenario *A*, the adversary controls the communication between the two devices on each side of the link. The adversary can influence both devices in a harmful way if its presence is undetected. Alternatively, a specific device in the network can be chosen as a target. Scenario *B* highlights how an adversary can gain additional advantages, e.g., making its localization more complicated. If the adversary can perform several simultaneous MIM attacks such that it is able to control all incoming and outgoing traffic for one specific device, it simulates the situation of a compromised device through ARP poisoning. Depending on the topology this can have different levels of complexity. The most appealing target for the adversary is most likely a grandmaster clock, as through this clock it can influence all slave clocks connected to it. According to the adversary model, if a clock is compromised, the adversary is capable of creating new synchronization messages, as well as delaying or accelerating messages it forwards. The latter can be achieved by changing the schedule of the clock. In this paper it is assumed that the adversary only performs attacks targeting links, i.e., scenario *A*.

C. Consequences of the delay attack

In the two scenarios *A* and *B* described above, once the first stage of the attack has been performed, i.e., the link has been compromised, the attacker performs the same action in both cases: it introduces a delay in the synchronization messages. In this section, an analysis of the consequences of introducing delays on the time synchronization in the network is shown.

If we use the value of the offset as obtained in (10.5) to correct the slave clock we have:

$$t_S^{old} \rightarrow t_S^{new} + \sigma_{\mathrm{m}eas} \; . \tag{10.7}$$

Using (10.7) in (10.6) we obtain $t_S^{new} - t_{GM} = 0$, making it the best possible value for the offset. This is the value that we would obtain in a attack free situation, therefore henceforth it will be referred as σ_{af}

$$\sigma_{af} = \frac{1}{2}((t_2 - t_1) - (t_4 - t_3)) \; . \tag{10.8}$$

The value of the offset obtained above, assumes that the transmission delay is symmetric. However, the adversary in the attack that we are considering (Fig. 10.7) introduces an asymmetric delay, d_{adv} that affects the synchronization messages in the following way:

$$\begin{aligned} d_{GM \rightarrow S} &= d_0 + d_{adv} \\ d_{S \rightarrow GM} &= d_0 . \end{aligned} \tag{10.9}$$

Now the time-stamps collected by the slave through the synchronization protocol described in Section 10.3-B are:

$$\begin{aligned} t_1' &= t_1 \\ t_2' &= t_2 + d_{adv} \\ t_3' &= t_3 + d_{adv} \\ t_4' &= t_4 + d_{adv} . \end{aligned} \tag{10.10}$$

Substituting (10.10) to (10.5) and using (10.8), we obtain the value of the measured offset when there is a delay introduced by the adversary according to:

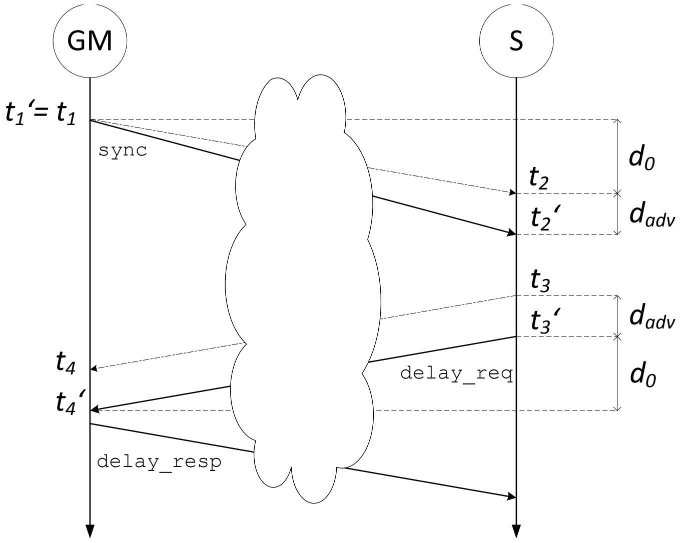

Fig. 10.7 Clock synchronization protocol under attack.

$$\sigma_{meas} = \sigma_{af} + \frac{1}{2} d_{adv} \ . \qquad (10.11)$$

If the adversary delays the delay request message instead of the synchronization message, only the sign of d_{adv} in (10.11) will change. The choice of message to delay does not affect the following reasoning. If σ_{meas} is used to correct the slave clock, then $t_S^{old} \rightarrow t_S^{new} + \sigma_{meas}$ and because initially t_S - t_{GM} = σ_{af}, the difference between the slave clock and the grandmaster is now:

$$t_S - t_{GM} = \sigma_{af} - \sigma_{meas} = \frac{1}{2} d_{adv} \ . \qquad (10.12)$$

Taking into account that this is the value just after the correction, from where the slave clock will start drifting again and thus for the next re-synchronization interval it will be:

$$t_S - t_{GM} = \frac{1}{2} d_{adv} + \sigma_{meas} \ . \qquad (10.13)$$

The worst case would be if $|\sigma_{meas}|=\sigma_{max}$, combining this result with the synchronization condition (10.2), we can conclude that an attack that introduces an asymmetric delay, no matter how small, can break the synchronization. However, this holds true only as long as the chosen σ_{max} for the network is actually the minimum value possible as expressed in (10.3). In order to make the network more resilient to possible attacks, we can relax this assumption and a longer value can be chosen. Thus, let σ_{rel} be the maximum allowed offset, the synchronization condition would be:

$$\left| t_{S_i} - t_{GM} \right| < \sigma_{rel} \; . \tag{10.14}$$

Then we obtain a relation between d_{adv}, σ_{max} and σ_{rel} that states that in order to break the time synchronization, an attacker has to introduce a delay twice as long as the difference between σ_{rel} and σ_{max} or, stated differently, a network can tolerate attacks that introduce a delay twice as long as the difference between σ_{rel} and σ_{max} before the time synchronization is broken, thus

$$\left| \frac{1}{2} d_{adv} + \sigma_{\max} \right| < \sigma_{rel} \; . \tag{10.15}$$

According to the defined adversary model, the adversary strives to keep the network penetration unnoticeable. Furthermore, in some cases, the adversary can succeed in keeping the slave ignorant of the synchronization breaking whenever protection techniques are lacking. The best scenario for the adversary is to break the clock synchronization, while letting the slave think that it still is in a synchronized state. If as a result of the attack, the slave still thinks that (10.14) holds for him, but in reality the offset between the grand master and the slave is bigger than σ_{rel}, the adversary has succeeded. The advantage, from the adversary point of view, of such an outcome is that the system remains oblivious of its failure. This means that the system will not apply any countermeasures to mitigate the consequences and consequently will not be able to return to a safe state.

10.6. Potential Solutions and Mitigation Techniques

In this paper we propose the use of a Configuration Agent to detect network penetration via a MIM attack as described above. The idea is that the Configuration Agent will be able to detect the traffic anomalies associated with the attack, and use these to diagnose what is happening.

In addition, we have identified some mitigation techniques that can be used alone or in conjunction with the Configuration Agent to strengthen the

IEEE 1588 against delay attacks. These mitigation techniques are not enough by themselves to prevent, protect against or detect an attack, but they can be used to put some boundaries to the damage caused by the attack, thus increasing the resilience of the system.

A. Attack detection

The synchronization messages are sent from the grandmaster with a period equal to the re-synchronization interval, R_{int} and the use of transparent clocks eliminates any possible interference of the rest of the traffic in the network. This means that any variation in R_{int} of the synchronization messages as perceived in the slave could be a hint of something happening in the network.

To detect these anomalies, a Monitor should be placed in the slaves. There the Monitor will be tracking the arrival times of synchronization messages to the slave. The interarrival time between two consecutive messages is:

$$\Delta t_i = t_{i+1} - t_i \,. \tag{10.16}$$

Although, the synchronization messages are sent periodically, some variations of the interarrival time should be expected. Nevertheless, an abrupt and sudden change in the interarrival time could be an indication of an attack happening.

Of course, other subtler, smarter attacks are likely not to be detected just by inspection of the interarrival times. For those, we should use some previous knowledge of the network. Here, we will assume that the Configuration Agent has been active in the network for some time before the attack starts and thus we have a history of the arrival times of synchronization messages to the slave. With a set of n values we calculate the average of the interarrival time, that would be the re-synchronization interval, R_{int}, as perceived by the slave:

$$R_{\text{int}} = \frac{\sum_{i=1}^{n-1}(t_{i+1} - t_i)}{n-1} = \frac{t_n - t_1}{n-1} \,. \tag{10.17}$$

The main difference between using the interarrival times or the re-synchronization interval as a parameter to detect an attack is that the first is an instantaneous measure that shows right away if something is happening but a smarter attack can go undetected. On the other hand, using the re-synchronization interval it is possible to spot more subtle attacks but some data need to be accumulated before a trend arises.

B. Mitigation techniques

1) *Bounding the breach:* Recall that the IEEE 1588 clock synchronization protocol is based on the exchange of messages between the grandmaster and the slaves. Concretely, in Fig. 10.2 it can be seen how the message sync is sent at t_1 from the grandmaster to the slave and, as a response, the message `delay_req` is sent from the slave at t_3, arriving to the grandmaster at t_4. Simi-

larly, the slave is waiting the response from the grandmaster, the `delay_resp` message that arrives at t_5. These two request-response relations can be used to prevent that delay attacks take the clock in the slave irreversible away from the grandmaster clock. To do so, we define t_{ret} for the grandmaster and the slaves as the timespan between sending the message and receiving the corresponding response message:

$$\begin{aligned} t_{ret}^{GM} &= t_4 - t_1, \\ t_{ret}^{S} &= t_5 - t_3. \end{aligned} \tag{10.18}$$

The minimum value for these is the transmission time of the message in the best case, i.e., when it does not suffer any intervention in terms of delay attacks, queuing delays or MAC layer contention. Thus, let n be the number of hops that the message goes through, then the value of t_{ret} can be calculated as

$$\min(t_{ret}) = \frac{messageLength}{dataRate} \times 2n\ . \tag{10.19}$$

To calculate the maximum t_{ret}, contention and execution times in the nodes must be taken into account. For the contention we assume that the message can be delayed by at most one message of maximum length in each hop. For the execution time we assume the worst case execution time (t_{WCET}):

$$\max(t_{ret}) = \min(t_{ret}) + \frac{messageLength}{dataRate} \times 2n + t_{WCET} \tag{10.20}$$

In a small network as the one depicted in Fig. 10.3 with just four hops between the grandmaster and the slave and keeping aside the execution time, the range of t_{ret} is (8, 104) μs (assuming synchronization protocol messages of 126 bytes, a *dataRate* of 100 Mbps and for the contention using the maximum length for an Ethernet message, 1522 bytes). This means that any value that exceeds that range can imply that the network is under attack. Note however, that with this method only the delay attacks that introduce a delay longer than *max(t$_{ret}$)* are detected each time. An attack that introduces a delay of, e.g., 50 μs will not be detected in a situation of low contention even though it is clear from Section 10.7-B that this is a delay large enough to break synchronization. Hence, this method can not be used alone as a detection mechanism, but only as a mitigation technique to prevent the attack from causing excessive clock drifts.

2) *Relaxed Mode*: One of the possible network reactions to the detection of an attack is to switch to a relaxed synchronization condition mode. This means that σ_{max} in (10.2) is increased. This relaxed mode leads to degradation of the network service quality, but may enable fast network recovery. Obviously, the applicability of such an approach depends on the criticality level of the application and the estimated time needed for recovery. It should be mentioned that the ability of the system to switch to the relaxed synchroni-

zation condition mode should be considered already during the system development phase.

3) *Using environmental conditions:* IEEE 1588 targets industrial applications that implies coping with related environmental conditions (e.g., temperature, humidity). These conditions can influence hardware and particularly the clock crystals. To investigate possible consequences for clock synchronization, the message exchange between a grand master *GM* and a slave *S* through a set of routers $R_1, \ldots , R_n$ is considered, Fig. 10.8. At each message exchange chain, an error δ, that is caused by hardware time-stamping inaccuracy, is also considered.

For simplicity first we consider a synchronization message exchange without intermediate nodes, routes, as it depicted in Fig. 10.7. In this case, in order to take into account additional deviations caused by environmental fluctuations, the following substitutions are required:

$$\begin{aligned} t_1'' &= t_1 + \delta_1 \\ t_2'' &= t_2 + \delta_2 \\ t_3'' &= t_3 + \delta_3 \\ t_4'' &= t_4 + \delta_4 . \end{aligned} \tag{10.21}$$

Then by using (5) we can see that the resulting value of the new offset $\sigma_{meas-envir}$ can be obtained as:

$$\sigma_{meas-envir} = \sigma_{af} + (\delta_2 - \delta_1 + \delta_3 - \delta_4)/2 \ . \tag{10.22}$$

The values constituting the error δ_i can be grouped according to which node they are produced. Actually, errors made by the same node are similar if we consider events occurring close in time, implying that the events of sending the sync and the `delay_req` messages are likely to be subject to the same environmental delay, and conversely, the event of the receiving the sync and the delay requests messages are likely subject to the same delay, according to

$$\begin{aligned} \delta_{GM} &= (\delta_1 + \delta_4)/2, \\ \delta_S &= (\delta_2 + \delta_3)/2. \end{aligned} \tag{10.23}$$

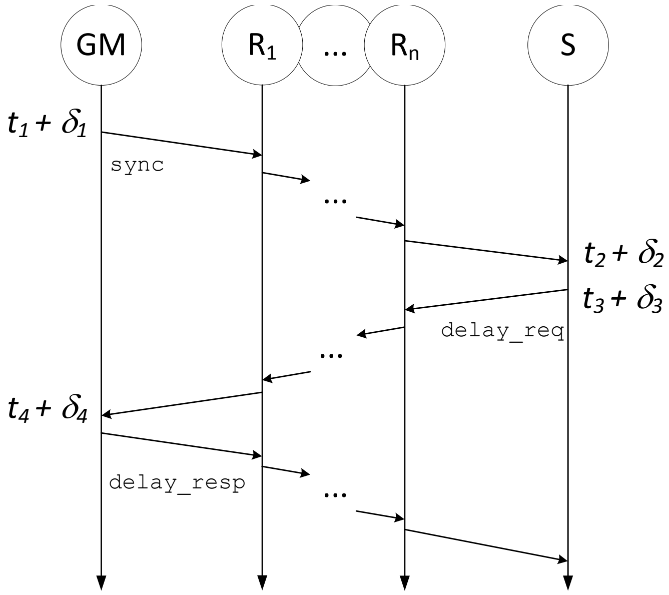

Fig. 10.8 Synchronization protocol considering additional clock drifts introduced by environmental conditions.

This yields:

$$\sigma_{meas-envir} = \sigma_{af} + \delta_S - \delta_{GM} \,. \tag{10.24}$$

If we add intermediate nodes, routers to the chain of synchronization messages exchange, their corresponding errors will be included in equation (10.24) twice (once per link they are connected to) but with difference signs. Each intermediate node is an end of the first related link and a beginning of the second related link, i.e., as we consider the difference of these errors for each link, they will have different signs in the final expression. Strictly speaking, when we subtract the two errors associated with the same node, the result is not exactly zero, but it is negligibly small. Therefore, when considering the errors caused by environmental conditions only the grandmaster and the slave errors are significant even if intermediate routes are included in the path. This leads to the conclusion that the most significant influence from the environment occurs when the grand master and the slave are separated far enough, such that they can have different environmental conditions.

This knowledge can also be used for detecting abnormal delays in the communication that cannot be explained by the environmental conditions alone. If nodes have sensors collecting data about main factors influencing clocks crystals, it is viable to calculate possible clock offset between nodes caused by different environmental conditions. If the observed shift is bigger than what can be expected by environmental conditions, this can indicate the presence of an adversary. Having a clock offset bigger than what was estimated can trigger additional checking of, e.g., the links for asymmetry delay detection. Under the assumption of having 5 μs offset with 50 ppm drift, environmental conditions can add 10 ppm to the drift and result in 6 μs offset [19]. This number shows that if the clocks are without temperature compensation, they can affect clock synchronization quite significantly.

C. Countermeasures discussion

When an attack is detected, while the system is in the Relaxed Mode, it should first, try to mitigate existing consequences, i.e., the synchronization breaching and, second, try to prevent the propagation of further consequences. To complete the first goal, related network participants should be informed that there are compromised links. Once the attack is detected, the Monitor could simply indicate between which grandmaster and which slave that there is a breach. The Monitor typically knows the path on which this breach occurred, but it is not known where exactly the adversary is. In the worst case, the whole path from the considered grandmaster to the slave should be eliminated from the clock drift calculations. In Scenario *A*, the adversary localization can be made by checking, one by one, all the links in the path under the assumption that there is a technique for checking the suspicious link without letting the adversary know about the check. It can be forged synchronization messages, where delaying would directly reveal the presence of the adversary. It is a challenge, as there are many parameters to consider and assumptions to validate. In Scenario *B*, additional measures should be applied for the adversary localization. In this case, the adversary can act on different links in different order. The possibility of this scenario is as high as the first one, as the adversary does not need any additional techniques for switching from Scenario *A* to Scenario *B*. Such switching will bring only benefits to the adversary, as it increases the chances for a successful attack and a longer undiscovered period which in turn means more serious consequences for the network.

10.7. Results

A. Attack evaluation

In this subsection, an evaluation of the attack targeting clock synchronization is presented. The attack consists of two phases. The first phase is a MIM attack via ARP poisoning and the second phase is a delay attack. The first phase is evaluated by formal specification of the conducted attack, and the second phase is evaluated by means of logical reasoning in the subsection 10.4-C. ARP poisoning is not a newly discovered type of the attack and the evaluation made in this paper is an extension of [4], which only considered Scenario *A*. We conduct a formal evaluation of this attack keeping in mind future work where we need a tool for investigating and comparison mitigation and prevention techniques.

For the formal attack description and evaluation, the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool was used. This tool is used for protocols analysis from a security point of view. It has several possible techniques for evaluation of security properties of the considered protocol, they are the On-the-fly Model Checker (OFMC), the Constraint-Logic-based Attack Searcher (CLAtSe), the SAT-based Model Checker (SATMC) and the Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP).

The OFMC [20] mode was used in the evaluations, as it allows to include an adversary in the list of network participants and by specifying the goal in the correct way, to check the knowledge of all participants in the end of the message exchange.

AVISPA uses High-Level Protocol Specification language (HLPSL) to interact with a user, and there is also a possibility to use the Security Protocol Animator (SPAN) [21] tool to simplify this interaction. Working with SPAN, the user needs to specify several categories of protocol components: identifiers, messages, knowledge and goals. Below, the formalization of Scenario *B* is considered. Formal analysis of both scenarios was performed, but as Scenario *B* can be represented as an extended Scenario *A*, only the evaluation of Scenario *B* is described here.

Identifiers. Two types of identifiers were used, they are users and numbers (see Table 10.1). In Scenario B (Fig. 10.5) there are 4 users: R_D, R_2 and *S* are benign network participants and AD_B is an adversary. IP and MAC addresses of these four users are represented as numbers.

Messages. ARP requests and responses are specified through messages. Intruder AD_B sends a message (ARP request) to R_D, this message contains the IP address of R_D and any other IP address of a benign networks participant. The node R_2 answers to *S* with a message that contains the MAC address of R_2. After such a manipulation, the adversary obtains the MAC address of R_2. In a similar manner, AD_B obtains the MAC addresses of R_D and *S*, now AD_B can send messages to all of them. In the next step, AD_B sends to R_D a message (ARP reply) containing the IP addresses of R_D and R_2, plus the MAC address of R_2. After this, for R_D the IP address of R_2 is associated with the MAC address of AD_B. This procedure needs to be repeated for R_2 and *S*. As a result, R_2

has both the IP addresses R_D and S associated with the MAC address AD_B and S has the IP address R_2 associated with the MAC address AD_B.

Knowledge. Each participant knows its IP and MAC addresses and other benign network participants (see Table 10.2).

Goals. The goal was specified as keeping the MAC address of R_2 secret from R_D and S. If this condition is fulfilled, it means that the attack was performed successfully, as R_D and S possess only the MAC address of AD_B while they think that they communicate with R_2. Therefore, if the tool shows that the secret holds, this means that the adversary wins.

Table 10.1 Identifier Declaration

Type	Identifier
User	R_D, R_2, S, AD_B
Number	IP_{R_D} , IP_{R_2} , IP_S, MAC_{R_D} , MAC_{R_2} , MAC_S, MAC_{AD_B}

Table 10.2 Identifier Declaration

User	Knowledge
R_D	R_2, S, IP_{R_D} , MAC_{R_D}
R_2	R_D, S, IP_{R_2} , MAC_{R_2}
S	R_D, R_2, IP_S, MAC_S
AD_B	R_D, R_2, S, IP_{R_D} , IP_{R_2} , IP_S, MAC_S, MAC_{AD_B}

OFMC analysis showed that specified in this way, the protocol is safe for the goal of keeping the MAC address of R_2 secret. This means that the described attack scenario is possible and can be performed. The second step is performing the delay attack. After successfully performing the MIM attack, R_D and R_2 communicate through AD_B, and R_2 and S communicate through AD_B. This means that AD_B can delay selective messages in these two communication channels. As it was shown in Section 10.4 this can lead to clock synchronization breaking.

B. Simulations with OMNeT++

To evaluate the proposed approach, we have created a network simulation using OMNeT++ [7] together with the INET framework [22]. For the concrete modules needed for the simulation of the clock synchronization protocol, the implementation made by Lévesque et al has been used [23]. The goal of these simulations is first to demonstrate that the delay attack can indeed break the clock synchronization. The simulations will also fulfill the role of the Monitor as part of the data that we obtain from them is the same data that a real Monitor would gather in a real network.

Fig. 10.3 shows the topology of the simulated network. The communication starting in the grand master *GM* and going to the slave *S* through the downstream router R_D. Communication starting in the slave goes to the grandmaster through the upstream router R_U. The re-synchronization interval R_{int} is set to 100 µs. We assume that the drift rate of the slave clock is 50 ppm and, therefore, applying (10.3), $\sigma_{max} = 10$ μs. And we chose $\sigma_{rel} = 20$ μs, thus implying that the system has been designed to work properly with this synchronization accuracy. Without loss of generality, just in order to simplify the explanations of the simulations, we assume that the master has a perfect clock, i.e., it does not drift. However, the slave is, of course, not aware of this fact.

We simulate the effects of an attack that breaks the time synchronization as shown in Section 10.5-C. For that we use different models for the delay: a constant delay and a linearly increasing delay. Once an adversary penetrated the network, it can use different techniques for messages delaying. The goal is to investigate different cases going from the simplest one to more complex and try to analize the differences from the detection point of view.

1) *Constant delay,* $d_{adv} = 50$ μs*:* In Fig. 10.9 the variations of the difference between the slave clock and the grandmaster clock with time is shown. Before the attack the difference was oscillating between 0 and -5 μs, as the result of the clock synchronization protocol performance. After the attack, the difference grows and oscillate between 25 μs and 30 μs. Because these values are bigger than σ_{rel}, we conclude that this attack breaks the clock synchronization. This is the expected result as the value chosen for the delay satisfies (15).

Although Fig. 10.9 is useful to show how the time synchronization is broken, it can be obtained just in the context of this simulations and not in a real-life situation. To detect the attack we must restrain the information used to the one available to the slave.

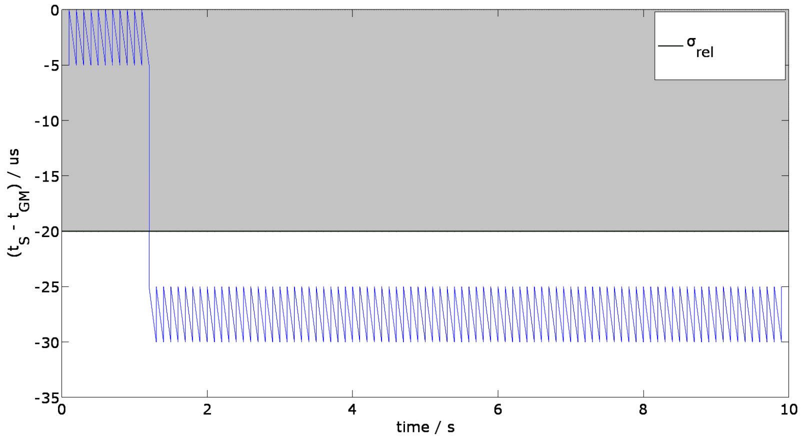

Fig. 10.9 Time deviation between the grand master clock and the slave clock. Constant delay, d_{adv} = *50 μs.*

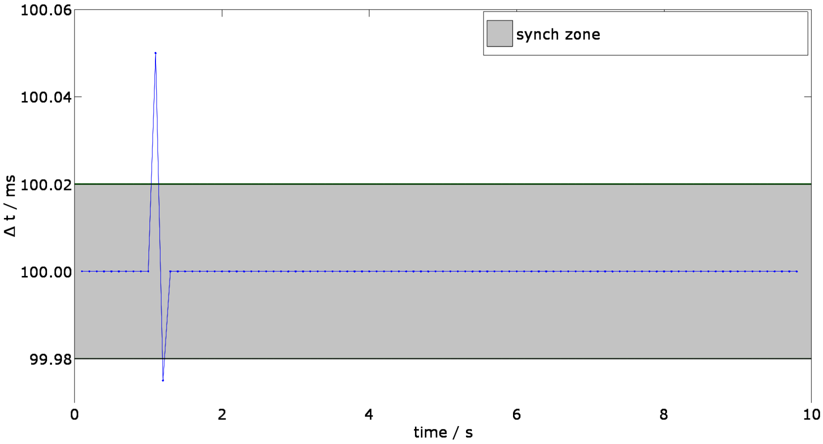

Fig. 10.10 Interarrival times of sync messages to the slave. Constant delay, d_{adv} = *50 μs.*

As it was explained in Section 10.4-A, the Monitor in the slave collects the arrival times of synchronization messages. Fig. 10.10 shows the interarrival times of synchronization messages to the slave as obtained with (10.16). Before the attack the interarrival times were constantly equal to the resynchronization interval. The peek in that figure is the first message affected by the delay. However, all the following messages are also affected by the delay, we can not see it in the figure because the delay is constant, therefore it only shifts the arrival time of the messages, but not the distance between them.

The value of the peak in Fig. 10.10 ($d_{adv} = 50\ \mu s$) is longer than the maximum possible value ($2\sigma_{max} = 20\ \mu s$), therefore, this attack will be easily detected just by analyzing the interarrival times.

After the attack has been detected, some mitigation techniques can be applied. For example, if the system has been designed to function in a relaxed mode such as $\sigma_{rel} > 30\ \mu s$ then the Configuration Agent can carry out this change of mode. Thus, even though the synchronization is deteriorated the system still behave in a predictable manner.

2) *Linearly increasing delay:* We now simulate a delay that increases linearly with every synchronization message that arrives to the router:

$$d_n = d_{adv} n\ , \qquad (10.25)$$

where $d_{adv} = 1\ \mu s$ and $n = 1; 2; \ldots$ for all messages after the attack starts.

In Fig. 10.11 it can be seen how the initial delay $d(1) = d_{adv}$ is not big enough to break the synchronization, but after enough re-synchronization intervals it does. However, in this case, as compared to the previous one, the attack cannot be detected just by inspecting the interarrival times of synchronization messages to the slave. This can be seen in Fig. 10.12: the effect of the attack on the interarrival times is so small that the slave might as well confuse it with a drifting in the grandmaster clock. This attack puts the slave in a state in which it is not aware of the fact that it is going out of synchronization when, indeed, it is.

For this case we conclude that other parameters, different than the interarrival times, should be used to be able to detect the attack. If we want to keep the detection local to the slave, we can assume that the Monitor has been gathering data for some time before the attack happens and use those values to obtain statistical parameters.

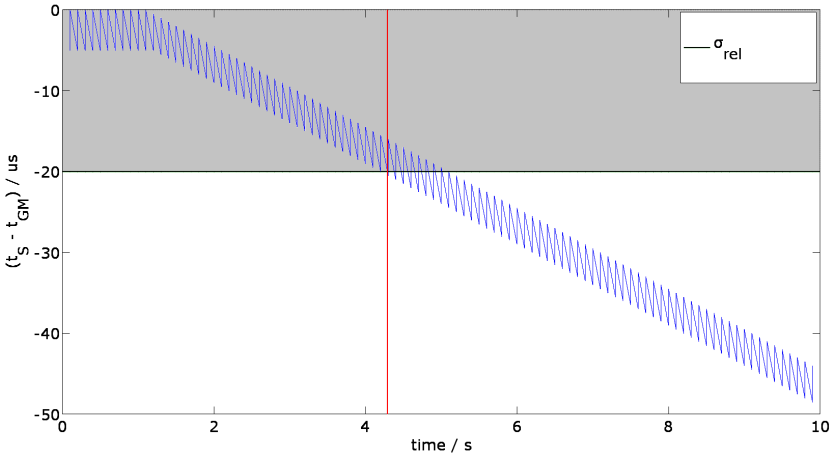

Fig. 10.11 Time deviation between the grand master clock and the slave clock. Linearly increasing delay.

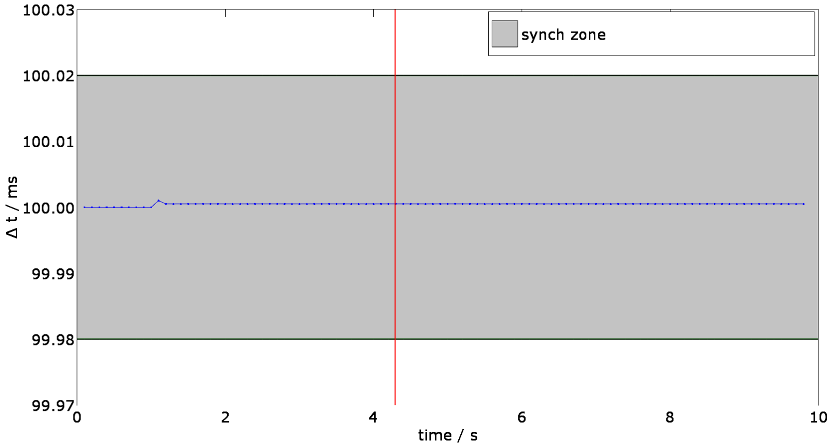

Fig. 10.12 Interarrival times of sync messages to the slave. Linearly increasing delay.

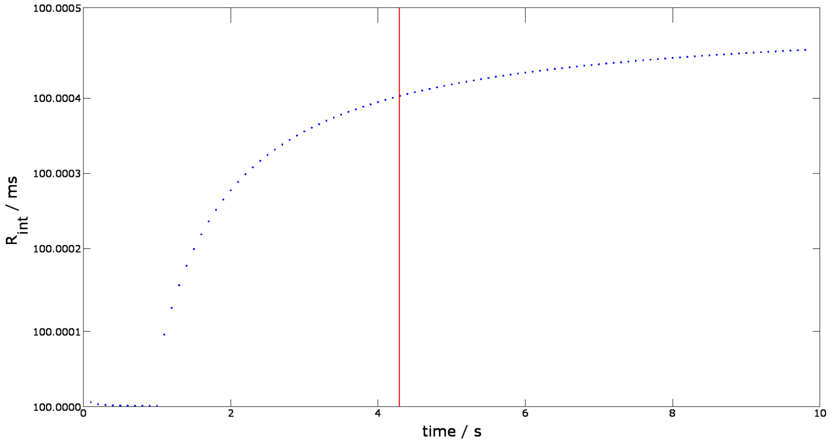

Fig. 10.13 Re-synchronization interval as obtained in the slave. Linearly increasing delay.

For example, we can examine the variations of the calculated value of the re-synchronization interval obtained using (10.17). In Fig. 10.13 it is shown how this value is consistently increasing. Therefore, in order to detect the attack we need to be able to identify this kind of patterns. Because the increase on the value is so small, we could probably not use it a sole criterion to detect an attack, but it can be one of a multi-criteria detection method. A further study could be done comparing the amount of re-synchronization intervals that the attack will need to break the time synchronization with the number of resynchronization intervals that the Configuration Agent needs to detect the attack.

Independently of the detection capabilities, here we see again how choosing a σ_{rel} longer than σ_{max} gives the network some time to react to the attack even before the synchronization has been broken.

We showed the inner difficulty of attack detection only by means of local monitoring, especially in the case of a smarter attack that introduces a linearly increasing delay. Our proposal is to use mitigation techniques, e.g., like the ones presented in Section 10.6-B, to cope with attacks that cannot be detected by distributed monitoring in nodes. If we apply the mitigation technique described in Section 10.6-B.1 to the topology used in the simulations we see that in the worst case scenario (i.e., low contention in the network) this method is able to detect attacks introducing delays longer than 100 μs. This value is way above the minimum delay required to break the synchronization and therefore, the technique cannot be used to prevent time synchronization breaching. Nevertheless, it has two important benefits in the case of the linearly increasing delay. First, it actually allows the slave to detect an attack that it is not possible to detect with the monitoring approach. And secondly, by including the knowledge of this upper limit for the delay in the design, the system can be prepared for this scenario, i.e., has an approach for returning into safe mode.

10.8. Conclusions and Future Work

In this paper, possible strategies of breaching clock synchronization together with techniques on how to detect it were investigated. First, the possibility of conducting the proposed attack breaching clock synchronization was proven by evaluation in AVISPA and through logical reasoning. This conclusion demonstrates the necessity to provide industrial networks with appropriate protection measures. Next a traffic monitoring approach was proposed as a mean of detecting the delay attack. Simulation of the Monitor analysis showed the possibility to detect the delay attack in the case of imposing constant delay. Simulation results also showed that if the system is designed with a relaxed synchronization condition mode, it can help with mitigating the consequences of a delay attack once it has been detected. The localization of the adversary depends on the way of performing the ARP poisoning attack, e.g., in the scenario, when an adversary takes over control of several communication channels, it is more challenging to define which links are compromised. Furthermore, the results also demonstrated that in the case of linearly increasing delay adversary influence can remain undetected. Therefore, more sophisticated detecting techniques are needed to detect such attacks. Algorithms for growing trend detection can then be a possible solution for coping with non-constant delays.

Clock synchronization is an essential part of all networks with real-time requirements. Basically all industrial networks have real-time requirements, and thus if there is a way to breach clock synchronization, the method will disrupt the network functionality, and is applicable for a range of use cases. IEEE 1588 is typically used to provide clock synchronization, but lacks security mechanism. Not even the Annex K, which has been introduced to enhance security, is capable of handling delay attacks such as the ones evaluated in this paper. However, there is nothing in IEEE 1588 which prevents us from using a monitor and thus our proposed solution can be used to enhance the standard.

There is a high potential for future work in this area. We plan to consider more attack scenarios, which include compromised devices and cases with clock acceleration and deceleration. Further, different detection and mitigation strategies, such as distributed monitoring to help locating the adversary together with algorithms for trend detection are to be considered. Furthermore, we want to add learning and adaptation abilities of an adversary and the Monitor to analyze their interactions.

Acknowledgments

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n°607727.

Bibliography

[1] D. Dzung, M. Naedele, T. von Hoff, and M. Crevatin, "Security for Industrial Communication Systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1152–1177, June 2005.

[2] E. Lisova, E. Uhlemann, J. Akerberg, and M. Bojrkman, "Towards secure wireless TTEthernet for industrial process automation applications," in *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, Sept 2014, pp. 1–4.

[3] E. Lisova, E. Uhlemann, W. Steiner, J. Akerberg, and M. Bjorkman, "A survey of security frameworks suitable for distributed control systems," in *2015 IEEE International Conference on Computing and Network Communications (CoCoNet),* December 2015.

[4] ——, "Risk evaluation for clock synchronization from arp poisoning attack in industrial applications," in *IEEE International Conference onIndustrial Technology (ICIT2016)*, 2016.

[5] B. Mukherjee, L. Heberlein, and K. Levitt, "Network intrusion detection," IEEE Networks, vol. 8, no. 3, pp. 26–41, May 1994.

[6] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heam, O. Kouchnarenko, J. Mantovani, S. Modersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Vigano, and L. Vigneron, "The avispa tool for the automated validation of internet security protocols and applications," *in Proceedings of the 17th International Conference on Computer Aided Verification*, ser. CAV'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 281–285.

[7] "OMNeT++," http://www.omnetpp.org/, 21 January 2015.

[8] M. Ullmann and M. Vogeler, "Delay attacks - implication on ntp and ptp time synchronization," in *Precision Clock Synchronization for Measurement, Control and Communication,* 2009. ISPCS 2009. International Symposium on, Oct 2009, pp. 1–6.

[9] 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.

[10] T. Mizrahi, “A game theoretic analysis of delay attacks against time synchronization protocols,” in *2012 International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS),* Sept 2012, pp. 1–6.

[11] M. Gutiérrez, W. Steiner, R. Dobrin, and S. Punnekkat, “A configuration agent based on the time-triggered paradigm for real-time networks,” in *2015 IEEE World Conference on Factory Communication Systems (WFCS),* May 2015, pp. 1–4, Best Work-in-Progress Paper Award.

[12] ——, “Learning the parameters of periodic traffic based on network measurements,” in *2015 IEEE International Workshop on Measurements & Networking (M&N),* Oct 2015, pp. 1–6.

[13] L. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J. Wood, and D. Wolber, “A network security monitor,” in *Society Symposium on Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer,* May 1990, pp. 296–304.

[14] M. Eslahi, M. Rohmad, H. Nilsaz, M. Naseri, N. Tahir, and H. Hashim, “Periodicity classification of http traffic to detect http botnets,” in *2015 IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE)*, April 2015, pp. 119–123.

[15] M. Garuba, C. Liu, and D. Fraites, “Intrusion techniques: Comparative study of network intrusion detection systems,” *in Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*, April 2008, pp. 592–598.

[16] G. Fink, B. Chappell, T. Turner, and K. O’Donoghue, “A metricsbased approach to intrusion detection system evaluation for distributed real-time systems,” *in Parallel and Distributed Processing Symposium, Proceedings International, IPDPS* 2002, Abstracts and CD-ROM, April 2002, pp. 8 pp–.

[17] “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.

[18] B. Kang, P. Maynard, K. McLaughlin, S. Sezer, F. Andren, C. Seitl, F. Kupzog, and T. Strasser, “Investigating cyber-physical attacks against iec 61850 photovoltaic inverter installations,” in *Emerging Technologies Factory Automation (ETFA), 2015 IEEE 20th Conference on*, Sept 2015, pp. 1–8.

[19] TCXO, Temperature Compensated Crystal Oscillator. [Online]. Available: http://www.radio-electronics.com/info/data/crystals/tcxo.php [20] D. Basin, S. Modersheim, and L. Vigano, "Ofmc: A symbolic modelchecker for security protocols," 2004.

[21] Y. Glouche, T. Genet, O. Heen, and O. Courtay, "A security protocol animator tool for avispa," in *In ARTIST-2 workshop*, 2006.

[22] "INET Framework," https://inet.omnetpp.org/, 21 June 2015.

[23] M. Levesque and D. Tipper, "ptp++: A Precision Time Protocol Simulation Model for OMNeT++ / INET," *CoRR*, vol. abs/1509.03169, 2015. [Online]. Available: http://arxiv.org/abs/1509.03169

Chapter 11

Paper E: Game Theory Applied to Secure Clock Synchronization with IEEE 1588

Elena Lisova, Elisabeth Uhlemann, Wilfried Steiner, Johan Åkerberg, and Mats Björkman.

MRTC technical report, MDH-MRTC-309/2016-1-SE, Mälardalen Real-Time Research Center, Mälardelen University, March 2016 (a shortened version of this report is in submission to IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, Stockholm, Sweden, September 2016).

Abstract

Industrial applications usually have real-time requirements or high precision timing demands. For such applications, clock synchronization is one of the main assets that needs to be protected against malicious attacks. To provide sufficient accuracy for distributed time-critical applications, appropriate techniques for preventing or mitigating delay attacks that breach clock synchronization are needed. In this paper, we apply game theory to investigate possible strategies of an adversary, performing attacks targeting clock synchronization on the one hand, and a network monitor, aiming to detect anomalies introduced by the adversary on the other. We investigate the interconnection of payoffs for both sides and propose the quarantine mode as a mitigation technique. Delay attacks with constant, linearly increasing, and randomly introduced delays are considered, and we show how the adversary strategy can be estimated by evaluating the detection coefficient, giving the network monitor the possibility to deploy appropriate protection techniques.

11.1. Introduction

In industrial applications information usually has its validity time, after which it loses its value. This implies that messages shall meet their deadlines, and therefore, some kind of schedule must be followed. Consequently, nodes must share the same notion of time, i.e., the difference between the clocks within two nodes should be within the allowed boundaries, or in other words, the nodes should be synchronized [1]. Clock synchronization is therefore one of the main assets of any system with real-time requirements [2, 3].

Disrupting clock synchronization is thus an appealing target for an adversary as breaching it will affect the whole network. Moreover, in many networks, the same algorithms for clock synchronization are used, which means that a successful attack can be reused for completely different applications. There are several standards for providing and maintaining clock synchronization in industrial networks. The IEEE 1588 standard [4] is widely used as it allows keeping good precision and eliminate delays caused by processing time in intermediate nodes by using transparent clocks. From Annex K 2008 some additional security measures has been added [5]. However, these measures are not enough, as they cannot prevent the breaching of clock synchronization by a selective delay attack [6, 7]. One way to breach clock synchronization was proposed in [3], namely, a combination of an Address Resolution Protocol (ARP) poisoning attack followed by a consecutive selective delay attack.

In this paper, game theory is applied to investigate and formalize the adversary-network interaction considering clock synchronization protection issues. Game theory is a mathematical theory describing possible interactions and/or cooperation between rational actors and studies the decision-making process along with the corresponding outcomes. In this context, a game is a model of such process in which only the desired setting can be investigated so that the consideration is limited to a specific set of targeted conditions and requirements [8]. Game theory allows considering interactions of players with contradicting interests. This is usually the case in security, as the adversary and the network have opposite targets. Some main definitions and types of games applicable to network security are presented in [9]. Games can be cooperative or non-cooperative depending on the targets of the players, static or dynamic depending on the number of interaction rounds for the players etc. Game theory applied to an Intrusion Detection Systems is presented [10], where the authors investigate how this technique can be used for formal decision making, and theoretically derived a Nash equilibrium, which was used to analyze the specified game. In [11], game theory was used to address security in vehicular communications, where the authors target developing an optimized strategy against an adversary in different scenarios.

In [12] game theory is applied to analyze the influence of a delay attack on the Network Time protocol (NTP), which is used in IEEE 1588. The author considers two strategies for a node, it can "pass" or "drop" a synchronization packet. However, the presented game consists of one interaction between an adversary and a node making decision. In this paper, we consider three possible strategies for a network monitor, two of them are logically equivalents of "pass" and "drop" respectively, but the third one is called quarantine and allows a system to check the link and determine whether it is under attack and/or employ mitigation techniques. We also consider multiple interaction games that allows us to consider different ways of imposing delays. The author of [12] proposes a multipath data collection procedure as a prevention technique against delay attacks. We, in turn, concentrate our attention on monitoring techniques as a way to secure clock synchronization, although multipath data spreading can be also used in the considered industrial networks as a way to provide the desired level of reliability and availability. In addition, we propose a game theory framework allowing comparison and evaluation of different types of attacks targeting clock synchronization, namely, constant, linearly increasing and random delays. The main contribution of the paper is a formal analysis of the interactions between an adversary, attacking clock synchronization and a network monitor, proving network protection. Players, their strategies and payoff functions are considered to define the game. Furthermore, networks states are considered along with a set of rules for switching between them as a reaction to adversary actions. Quarantine Mode is proposed as one of the system states allowing additional techniques for adversary detection. Finally, the detection coefficient is introduced as a metric for adversary exposure in the network. This metric is used to reason about the efficiency of different types of attacks and predict the adversary behavior.

The reminder of the paper is organized as follows. Section 11.2 introduces our system model and explains the idea of Quarantine mode. Next, the game is formulated in Section 11.3, whereas Section 11.4 presents the analysis of an adversary influence in the network. The game analysis and a comparison of attacks via detection coefficients are presented in Section 11.5. Finally, Section 11.6 concludes the paper.

11.2. System Model

In order to formalize the interaction between an adversary and the monitor, the considered system model along with the participants of the game should be set.

A. Network Model

We assume that IEEE 1588 is used in the network for establishing and maintaining clock synchronization. This means that clock synchronization is achieved through message exchange between a grandmaster and a slave. First, the grandmaster sends out a time stamped `sync` message, and when the slave receives the message, it also timestamps it to determine the arrival time and sends out a `delay_req` message containing the two previous timestamps plus a new one to indicate the transmission time. Finally, when the grandmaster receives the message, it timestamps it to determine its arrival and sends out a `delay_resp` message. In the end of such an exchange, both the grandmaster and the slave have time stamps of the `sync` and `delay_req` messages at the moments of transmitting and receiving. Knowing these four values and assuming absence of asymmetrical delays, the offset between the two clocks can be calculated.

We use a distribution analysis of the measured offset, σ_{meas}. The offset is measured by a slave according to IEEE 1588 and the monitor in each slave is saving the measured offset in every re-synchronization interval and calculate statistics based on it [13]. In this paper, we consider only two basic statistic parameters, namely mean and standard deviation. We assign thresholds for each of them, which we refer to as indicators. We say that an indicator is positive when it is above the allowed threshold and that it is negative otherwise. According to our assumption, the resulting offset measured by a slave consists of three components: offsets related to the clock drifts, related to a natural delay in the communication channel and, finally, related to the adversary. Offsets related to the clocks drift is always present therefore, without loss of generality, it can be excluded from the consideration. The clock drifts do not affect the overall reasoning and calculations, but changes only the threshold set for making a decision about switching to a different system mode. The considered network is heterogeneous, implying that is contains a mixture of wires and wired communication links, such that the route between a grandmaster and a slave can consist of both types of links. In wireless channels without fading, the variations in propagation delay were found to have an exponential distribution [14]. Hence, we model offsets related to nature with an exponential distribution, since wires point-to-point links likely experience smaller delay variations compared to line-of-sight-wireless links, and thus the worse-case scenario is considered.

B. Adversary Model

We assume that the adversary is using a combination of an ARP poisoning attack and a selective delay attack to break clock synchronization [3]. To be able to breach synchronization, the adversary does not need to forge or modify the synchronization messages, only to delay it. This is a reason why encryption cannot help against this type of attack, as the timestamps already incorporated in the message do not need to be modified.

C. Introducing Quarantine Mode

The idea of introducing Quarantine Mode is to be able to react to an attack before it breaches clock synchronization. If there is an indicator of any abnormal behavior, the system puts the suspicious link/route into quarantine or switches it to Quarantine Mode. In this mode, the link is still used, but the system simultaneously tries to find out whether the abnormal behavior was an error or a consequence of a malicious interference with the network. This can be achieved by e.g., using additional techniques to provoke the adversary in such a way that it reveals itself or/and by further monitoring the network/link characteristics. In the paper, we consider the second option, leaving the first one for future work. Therefore, in Quarantine Mode besides the two initials indicators, others can also be considered, e.g. like monitoring the maximum return time for the messages or checking the current environmental conditions [13]. The benefits of this mode includes the possibility to check what is going on with the link and try to mitigate any problems smoothly, while still continuing to do the best possible for clock synchronization.

In our previous work, the term Relaxed Mode was introduced [13]. Relaxed Mode implies a degraded quality of synchronization, but also gives the system the opportunity to recover to a safe state. Quarantine can be considered as an extension of the previously introduced Relaxed mode. Quarantine not only gives system an opportunity to recover if it is under attack, but also gives the opportunity to make a decision about whether there is an adversary in the network or not. Relaxed mode was introduced by using relaxed boundaries for the offset between nodes, and we introduced trust coefficient that can be considered as equivalents of relaxed boundaries for clock offsets.

Also it should be mentioned that the system can switch from normal working state to the state where the attack has been detected without entering Quarantine Mode, if the symptoms or indicators of being under attack have significant values or significantly many are positive simultaneously.

11.3. Proposed Game Model

The main components of a game are actors, their strategies and the payoffs functions. We complete this set with probabilistic functions for the game strategies, as this allows us to bind all components together and to analyze the possible adversary behavior and consequences for the network.

A. Actors

We consider three actors in this game. The first one is the adversary, which targets to breach clock synchronization and, in the best case, stays undetected. The second one is the network monitor in the node that wants to stay synchronized with the grandmaster. Note that we consider local detection only in this paper. Third player is the nature or the environment or the channel. It does not have strategies, but it has a probabilistic distribution of the delays in channel. These delays can affect clock synchronization or/and mask adversary actions.

B. Strategies

By strategy, we understand the set of possible delays that can be introduced by the players. The game strategy space can be presented as:

$$S = S_{nat} \times S_{adv} \times S_{mon} \qquad (11.1)$$

where × is Cartesian product, and S_{nat}, S_{adv} and S_{mon} are nature, an adversary and the monitor strategies respectively.

For an adversary, it is reasonable to impose a delay only in one direction in order to make the delay asymmetrical, since IEEE 1588 can be breach only by an asymmetrical delay [3]. Nature imposes delays in both directions, however for simplicity we consider also this delay as asymmetric since this is the most troublesome type of delay. Nature only has one strategy:

$$S_{nat} = \{d_{nat} \mid d_{nat} > 0\}, \qquad (11.2)$$

where d_{nat} is a delay caused by the nature.

We consider four different strategies for the adversary; no attack, introducing a constant delay, a linearly increasing delay or a random delay. Therefore, the space of the adversary strategies can be presented as

$$S_{adv} = \left\{S_{adv}^{NA}, S_{adv}^{CD}, S_{adv}^{ID}, S_{adv}^{RD}\right\} \ . \qquad (11.3)$$

The corresponding strategies are defined as:

- No Attack (NA), the related strategy is a set of possible delays, which consist of only one element – 0.

$$S_{adv}^{NA} = 0 \, . \tag{11.4}$$

- Constant Delay (CD)

$$S_{adv}^{CD} = \{ d_{adv} \mid d_{\min} \leq d_{adv} \leq d_{\max} \} \, , \tag{11.5}$$

where S_{adv}^{CD} is the strategy of imposing a constant delay, d_{min} and d_{max} are minimum respectively maximum values for the delay imposed by the adversary, d_{adv}.

- Linearly Increasing Delay (ID)

$$S_{adv}^{ID} = \left\{ i \cdot d_{adv} \middle| \, 0 < d_{adv} < d_{\max} \right\} , \tag{11.6}$$

where i is the number of the synchronization interval, i.e., an iteration for the adversary.

- Random Delay (RD)

$$S_{adv}^{RD} = \left\{ (d_{adv,i}, 0) \middle| \, 0 < d_{adv,i} < d_{\max} \right\} , \tag{11.7}$$

where $d_{adv,\, i}$ is a random delay imposed by the adversary at the i^{th} iteration.

For the monitor, three strategies are proposed, link/route is in quarantine, attack not detected, attack was detected, and thus the space of the monitor strategies can be presented as

$$S_{mon} = \left\{ S_{mon}^{AND}, S_{mon}^{Q}, S_{mon}^{AD} \right\} \, . \tag{11.8}$$

- *Link/Route is in Quarantine (Q)*, so the system switched to suspicious mode and can start applying additional techniques to confirm the attack.

$$S_{mon}^{Q} = \left\{ \sigma_{applied} = \sigma_{meas} \cdot \beta \;\middle|\; 0 < \beta < 1 \right\} , \tag{11.9}$$

where β is the trust coefficient, and it shows that in this mode it is reasonable to question the validity of the value of the calculated offset, σ_{meas} is the measured offset and $\sigma_{applied}$ the offset that is used for actual correction. The value of the trust coefficient can be connected to the boundaries allowed for two nodes to stay synchronized.

- *Attack Not Detected (AND)*, normal mode of functioning for the node. This situation can be described as the previous one but with β=1:

$$S_{mon}^{AND} = \left\{ \sigma_{applied} = \sigma_{meas} \right\} . \tag{11.10}$$

- *Attack was detected (AD)*, in this mode the system is assured that it is under attack, so it can start applying mitigation techniques. The trust in the calculated offset in this mode is β=0, as in this case the monitor does not trust in the calculated value at all:

$$S_{mon}^{AD} = \left\{ \sigma_{applied} = 0 \right\}. \tag{11.11}$$

C. Payoff Functions

In Table 11.I each cell represents a game state, that can be characterized by a specific combination of monitor and adversary strategies. Each cell consists of a pair of values: the first one is the payoff for adversary, and the second one is the payoff for the monitor or the network. A payoff is a positive or negative number that allows a comparison of different states of the game from every participant's point of view. In the table, C_k, is defined as a positive value, and the negative sign is used to indicate loss. We assume that the adversary and the monitor are rational, thus they want to move to a state with higher payoff.

Table 11.I. Payoffs for adversary and monitor strategies.

	AND	Q	AD
NA	(0, 0)	$(0, -C_1)$	$(0, -C_2)$
CD	$(C_7, -C_8)$	$(-C_3, C_6)$	$(-C_4, C_5)$
ID	$(C_7, -C_8)$	$(-C_3, C_6)$	$(-C_4, C_5)$
RD	$(C_9, -C_{10})$	$(-C_3, C_6)$	$(-C_4, C_5)$

To reason about the behavior of players and the relations between the payoffs, the following sets are used:

$C_3 < C_4$ - there is a chance for the adversary to stay undetected even in Quarantine Mode;

$C_1 < C_2$ – it is not critical to switch into quarantine while not being under attack, whereas it is critical to start mitigation techniques and possibly switching into safe mode while the system is under attack.

$C_2 < C_8$ - the worse-case scenario for the system is not to detect or start being suspicious only when there is an ongoing attack;

$C_5 \cong C_6$ - what is more valuable: to detect an attack in the end or to have the opportunity to start some additional measures while being in Quarantine Mode? This depends on the system assets and their cost. In our assumption considering safety-critical industrial applications, it is more important to detect an adversary in the network even if it requires implementation of additional techniques.

$C_{10} > C_8$ - the potential harm from a random delay attack is less for the system, as the value of the delay not necessary breaks synchronization, e.g., if it is small enough and does not bring the node out of its synchronization boundaries.

$C_9 < C_7$ - as the potential harm is less in case of random delays, consequently, the potential gain from its detection is also less, compare to the other cases.

D. Probability functions

The idea is to assign probabilities to all strategies or states. Probabilities for the adversary strategies, are not known initially, even though they can be estimated during the process of interaction, and are set as follows:

$$p_{NA} + p_{CD} + p_{ID} + p_{RD} = 1\,, \tag{11.12}$$

where p_{NA}, p_{CD}, p_{ID}, and p_{RD} are probability of NA, CD, ID, and RD modes respectively. One of the targets of the analysis can be these probabilities estimation, this can allow to predict adversary behavior.

For the monitor the probabilities can be calculated assuming that attacker's actions are known. As in our model choice of a strategy for the Monitor depends on the calculated values of offsets and its statistic characteristics, the probability of the choice can be calculated. Corresponding variables are:

$$p_{NAD} + p_Q + p_{AD} = 1\,, \tag{11.13}$$

where p_{NAD}, p_Q, p_{AD} are probability of the strategies NAD, Q, and AD respectively.

As a criteria for switching the monitor into Q or AD mode, the mean and standard deviation of the delay are used as indicators:

$$\bar{\sigma} = \frac{1}{N}\sum_{i=1}^{N} \sigma_{meas,i} \quad , \tag{11.14}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (\sigma_{meas,i} - \bar{\sigma})^2}{N}} \quad , \tag{11.15}$$

where N – is a number of currently considered resynchronization intervals.

In this game, we assume that the decision about switching the state in the monitor depends on several thresholds for mean and standard deviation of the calculated offset. Therefore, the probability of being in each state can be taken from the probabilities of the delay value being in a corresponding interval (Fig. 11.1). These probabilities we can calculate by knowing the distributions for nature and using appropriate variables for the adversary state probability. For each indicator, there are two thresholds: low and high, i.e., Σ_L and Σ_H are the low and high thresholds for the standard deviation whereas $\bar{\Sigma}_L$ and $\bar{\Sigma}_H$ are the low and high thresholds for the mean respectively.

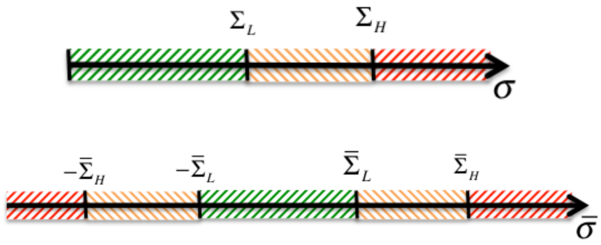

Fig. 11.1 Thresholds for the delay mean, $\bar{\sigma}$ and standard deviation, σ.

For each monitoring characteristic, we introduce two thresholds, low and high. If one of them is above the corresponding low threshold, but still lower than the high threshold, it is only suspicious. If the characteristic is above the high threshold, we assume that an attack is detected.

Switching rule:

- *Rule 1*. If one and only one of the monitoring characteristics is above the corresponding low threshold, the system is switched into the Q mode: $\bar{\sigma} > \bar{\Sigma}_L$ or $\sigma > \Sigma_L$.
- *Rule 2*. If two of the monitoring characteristics are above the low thresholds, the system is switched into the AD mode: $\bar{\sigma} > \bar{\Sigma}_L$ and $\sigma > \Sigma_L$.
- *Rule 3*. If any of the monitoring characteristics is above the corresponding high threshold, the system is switched into the AD mode: $\bar{\sigma} > \bar{\Sigma}_H$ or $\sigma > \Sigma_H$.
- *Rule 4*. If any of the cases described below is true, the system is switched from Q mode to AD mode:
    - If there are positive indicators of network anomaly from the additional checking techniques that were deployed as a result of switching to Quarantine mode;
    - If *Rule 1* or *Rule 3* can be applied.

11.4. Analysis of Adversary Influence

For influences by the communication channel we consider two cases: wired and wireless channels, but as both of them have the same distribution of the delays, the Probability Density Function (PDF) of channel delays (ChD) can be modeled by an exponential distribution:

$$f_{ChD}(d_{ChD}) = \lambda \cdot e^{-\lambda d_{ChD}},\ d_{ChD} \geq 0\ , \tag{11.16}$$

where d_{ChD} is a delay caused by the channel, and λ is a distribution parameter.

A. CD strategy

First we consider a constant delay imposed by the adversary, and assume that there is a discrete set D_{CD} of values with size N_1 from which the adversary can choose the imposed delay:

$$D_{CD} = \left\{ d_{CD,\,j} \right\}_{j=1}^{N_1}, \tag{11.17}$$

$$\sum_{j=1}^{N_1} p_{CD,\,j} = 1,\ p_{CD,\,j} = f_{CD}(d_{CD,\,j})\ , \tag{11.18}$$

where $p_{CD,j}$ is the probability that the adversary chooses delay $d_{CD,j}$ from the set to impose and f_{CD} is a discrete function relating a possible delay from the set with its possibility to be imposed.

The resulted offset measured by a slave will consist of these two delays: The PDF of two independent variables, $d_{CD,j}$ and d_{ChD}, is a convolution of the PDFs of these variables. Therefore, the PDF of the measured offset (without considering clock natural drifts) can be calculated as:

$$\begin{aligned} f_{offset}(\sigma_{meas}) &= \sum_{j=1}^{N_1} f_{ChD}(\sigma_{meas} - d_{CD,\,j}) f_{CD}(d_{CD,\,j}) = \\ &= \lambda e^{-\lambda \sigma_{meas}} \sum_{j=1}^{N_1} (e^{\lambda d_{CD},j} \cdot p_{CD,j}). \end{aligned} \tag{11.19}$$

The sum is a coefficient if the set of delays available for the adversary is fixed. Therefore, if we introduce the detection coefficient:

$$k_{CD} = \sum_{j=1}^{N_1} (e^{\lambda d_{CD,j}} \cdot p_{CD,j}), \tag{11.20}$$

we can see that, the PDF of the measured offset is proportional to the exponential distribution:

$$f_{offset}(\sigma_{meas}) = k_{CD} \lambda e^{-\lambda \sigma_{meas}} . \tag{11.21}$$

B. ID strategy

In the next case, we consider a linearly increasing delay imposed by the adversary. Here the adversary can choose the step of increment from a defined set with the size N_2. Thus, the imposed delay can be described as follows:

$$D_{ID} = \left\{ i \cdot d_{ID,j} \right\}_{j=1}^{N_2}, \tag{11.22}$$

$$\sum_{j=1}^{N_2} p_{ID,j} = 1,\ p_{ID,j} = f_{ID}(d_{ID,j}), \tag{11.23}$$

where i is the number of resynchronization intervals or the iteration of the delay attack. In this case, the PDF of the resulting measured offset can be calculated as:

$$f_{offset}(\sigma_{meas}) = k_{ID} \lambda e^{-\lambda \sigma_{meas}} , \tag{11.24}$$

$$k_{ID} = \sum_{j=1}^{N_2} (e^{\lambda \cdot i \cdot d_{ID,j}} \cdot p_{ID,j}) . \tag{11.25}$$

As we can see, it is again a scaled exponential distribution.

C. RD strategy

Finally, in the case of a random delay attack, we assume that the probability distribution is uniform:

$$f_{RD}(d_{RD}) = \frac{1}{d_{\max}},\ d_{RD} \in \left(0, d_{\max}\right]. \tag{11.26}$$

In this case, the PDF of the measured offset can be calculated as:

$$\begin{aligned} f_{offset}(\sigma_{meas}) &= \int_{-\infty}^{+\infty} f_{RD}(\sigma_{meas} - \tau) f_{ChD}(\tau) d\tau = \\ &= \frac{\lambda e^{-\lambda \sigma_{meas}}}{d_{\max}} \int_{0}^{d_{\max}} e^{\lambda \tau}\, d\tau = \frac{e^{-\lambda \sigma_{meas}}}{d_{\max}} (e^{\lambda d_{\max}} - 1) \end{aligned} \tag{11.27}$$

.

After introducing the corresponding coefficient for random delay attack, the PDF of the measured offset can be presented as an exponential distribution:

$$f_{offset}(\sigma_{meas}) = k_{RD}\lambda e^{-\lambda\sigma_{meas}}, \tag{11.28}$$

$$k_{RD} = \frac{1}{\lambda d_{\max}}(e^{\lambda d_{\max}} - 1). \tag{11.29}$$

For exponential distributions, the mean and deviation are well known, so given these PDFs we can define thresholds for system mode switching. The thresholds defined with knowledge about adversary behavior and possible attacks are ideal one that can allow the monitor to detect the attack. In reality we cannot obtain them, but nevertheless it is interesting to consider them to see possible ways of interactions between the adversary and the monitor as well as to play with values to investigate the limits for adversary detection.

For the monitor characteristic deviation, the lower threshold is the deviation of the corresponding exponential distribution, and higher thresholds can be chosen by taking the lower threshold and increasing it by γpercent. For the monitor characteristic mean, the mean of the exponential distribution sets only the middle of the allowed interval, so here we assume that the allowed deviation is γpercent for setting the lower threshold. For the higher threshold, we again set the allowed deviation to an additional γ percent of the lower threshold. Therefore, in the case of linearly increasing delay, the thresholds are:

$$\Sigma_L = \frac{k_{ID}}{\lambda^2},\ \Sigma_H = \frac{k_{ID}}{\lambda^2}(1+\gamma), \tag{11.30}$$

$$\bar{\Sigma}_L = \frac{k_{ID}}{\lambda}(1+\gamma),\ \bar{\Sigma}_H = \frac{k_{ID}}{\lambda}(1+2\gamma). \tag{11.31}$$

One interesting point here is that the obtained thresholds actually depend on the number of iterations, which is logical as the delay also depends on this number. For the case of constant or random delay the thresholds can be obtained by changing k_{ID} to k_{CD} or k_{RD} respectively.

11.5. Security Game

Having set a formal model of the game, we can try to play it by exploring possible interconnections and making numerical estimations of the detection coefficients for different adversary strategies.

A. Monitor State Machine

If the thresholds defined in the previous section are used for switching strategies in the monitor, the overall system becomes a state machine with probabilities of transmission between different states and remaining in the current one. Of course, this result is obtained under strong assumptions about the actions of the adversary, but it still shows that interaction between the adversary and the monitor can be described in a probabilistic manner. For deterministic networks, probabilistic characteristics are not sufficient enough, but they allow making intelligent predictions about adversary behavior and can be used to analyze the effects of prevention and mitigation techniques. Strategies for switching between modes can be compared based on the derived state machine configuration, where techniques with higher payoff functions are more efficient. Also this approach allows to consider a combination of techniques.

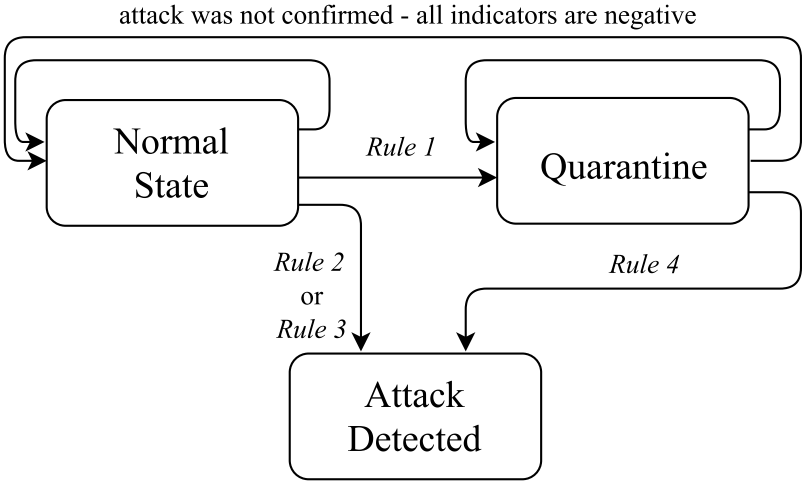

Fig. 11.2 States of the Monitor and transitions between them.

Fig. 11.2 shows the possible states of the monitor related to its corresponding strategies and their interconnections. We assume that after switching to Attack Detected state, the system deploys mitigation techniques, the details of which are not considered in this paper. The trigger for any transaction is a new calculated offset value for the current resynchronization interval. Based on such representation, reasoning about system stability can be conducted. If we can calculate probabilities for all transitions for each type of attack or we can compare these probabilities for different types of attacks, it can be used as an attack efficiency metric.

B. Can we make an adversary to lose a-priori?

It is also interesting to see how potential losses of both players are related. Looking at Table 11.I we can define the maximum gain for the adversary and the monitor respectively:

$$GainAdv_{\max} = C_7 \max\left\{p_{CD}, p_{ID}\right\} p_{AND}\,, \tag{11.32}$$

$$GainMon_{\max} = C_5 \max\left\{p_{CD}, p_{ID}\right\} p_{AD}\,. \tag{11.33}$$

Maximum loses for both sides can also be defined accordingly:

$$LossAdv_{\max} = C_4 \max\left\{p_{CD}, p_{ID}\right\} p_{AD}\,, \tag{11.34}$$

$$LossMon_{\max} = C_8 \max\left\{p_{CD}, p_{ID}\right\} p_{AND}\,. \tag{11.35}$$

As the goals of the monitor and the adversary are opposite, it is logical that the loss of one looks similar to the gain of the other. The exact relation depends on the ratio between C_4 and C_5, C_8 and C_7. If they are equal, then the game is a zero-sum game. It is an open question whether we can shift the ratio and make the adversary loose a-priory by applying system design choices, however the proposed game theory framework can be used as an evaluation tool to determine the applicability of each solution. Obviously, such design choices are the most expensive solutions as otherwise the ratio cannot be shifted, but then it can be approved for safety-critical applications.

C. Detection Coefficient

In case there is no adversary in the network, it can easily be seen that with the assumption of an exponential distribution of channel delays in equation (11.30-31) and with $k_{ID} = 1$ a reasonable threshold is one allowing the network to eliminate clock synchronization breaching caused by channel characteristics. Therefore by comparing the thresholds with and without considering an adversary present in the network, we can see that the adversary attack can be noticed when $k_{ID} > 1$. That is always the case, but the problem here is that this value can be very close to 1, so some qualitative metrics are needed to evaluate and compare different types of attacks. The difference between thresholds calculated with and without an adversary present, demonstrates the influence of the adversary. The bigger the difference, the more exposed the adversary is in the network. Therefore, if the coefficient is much bigger than 1 for an attack, this attack is not a good choice from an adversary point of view.

To compare the three types of considered delay attacks, we need to compare the values of the related detection coefficient from equations (11.20, 11.25 and 11.29) respectively, more precisely we need to compare their difference to 1, as the approach is more beneficial the more above the value 1 it is.

Therefore, for the considered types of attacks, we need to see how sensitive this coefficient value is. To this end, we compare the detection coefficients from the different types of attacks. However, we do not set probability correspondence between coefficient ratios and detection ratios. Fig. 11.3 demonstrates the detection coefficients and their dependencies of the value of imposed delays d_{CD} or d_{ID}, which are considered to be the same, to allow comparison of the CD and ID cases. These dependencies are obtained under the following assumptions:

- $N_1 = N_2 = 2$;
- $\lambda = 4$;
- $d_{CD,1} = d_{ID,1}$, $d_{CD,2} = d_{ID,2} = 20\mu s$
- $p_{CD,1} = p_{ID,1} = 0.4$, $p_{CD,2} = p_{ID,2} = 0.6$

Furthermore, several iterations are considered for ID and several values of d_{max} for RD. Fig. 11.3 demonstrates that for RD the level of being exposed in the network depends on the channel characteristics λ and d_{max} – it can be both less and more than the other two considered cases. Therefore, from an adversary point of view, the expediency of applying RD depends on the application specification: it is beneficial for the adversary if the network can be disrupted by a short-term breach of clock synchronization, especially in cases where bigger offsets between the nodes in the network is allowed, as this represents the right side of Fig. 11.3. An adversary employing CD or ID is more exposed in the network, the higher the value of the imposed delay. Based on the chosen metric of delay being mean and standard deviation, iterations with ID is more difficult to detect in the network, therefore, the question here is more how fast the breach can be detected.

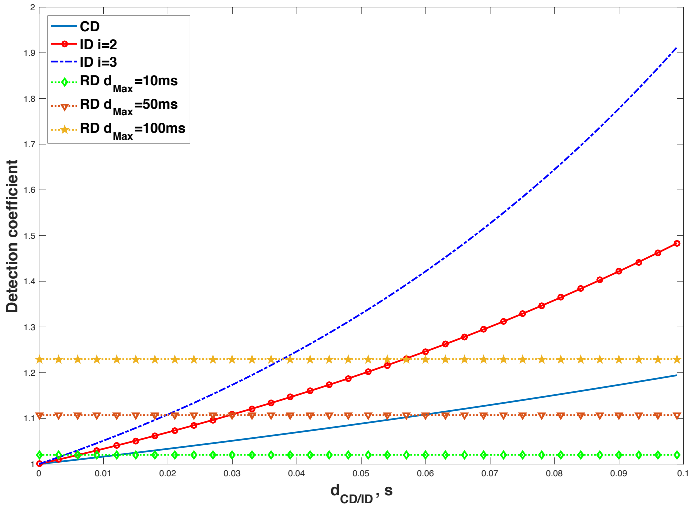

Fig. 11.3 Detection coefficients for RD, ID, and RD types of attacks.

The crossing points of the coefficient curves for different attack types are of special interest, as they show points where it is beneficial for adversary to make switch between different strategies. Generally speaking, the adversary wants to stay in the graph with as low detection coefficient as possible, as it indicates a lower exposure in the network. Thus, if from the very beginning, the adversary is applying the CD strategy (blue line without markers), it should switch to the RD strategy to minimize the exposure level at the moment when it crosses the RD line (yellow line with circular markers or orange line with triangular marker). Knowing this reasoning from an adversary point of view, however, is beneficial for the monitor as well. The monitor can predict and model possible adversary behavior and deploy proactive countermeasures given this knowledge. Therefore, if we can make valid assumptions about possible delays, this approach can be applied. The question is how to make those assumptions. One of the options if we have some history of interactions between the adversary and the monitor, is in case the monitor can learn the probabilities and possible delays and calculate the thresholds based on this. Therefore, a learning period can be used for collecting network statistic and during the interaction with the adversary, the monitor can learn its properties and react accordingly.

11.6. Conclusions and Future Work

In this paper, a formal analysis of the interaction between an adversary targeting clock synchronization via a selective asymmetric delay attack and a network monitor collecting statistics of measured offsets according to IEEE 1588 is presented. A game theory framework is proposed as an evaluation tool for the described interaction. The detection coefficient is identified as an appropriate evaluation metric for comparing different adversary strategies. Based on this approach, it is possible to say which strategy is most beneficial for the adversary depending on the set of game configurations. However, knowing this data, the monitor has a possibility to deploy appropriate protection techniques. In the future, we plan to include additional techniques that can be activated in Quarantine mode to help detecting an adversary.

Acknowledgments

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n°607727.

Bibliography

[1] Hermann Kopetz and Wilhelm Ochsenreiter, "Clock Synchronization in Distributed Real-Time Systems," IEEE Transactions on Computers, vol. C-36, no. 8, pp. 933-940, 1987.

[2] Wilfried Elmenreich, "Time-Triggered Fieldbus Networks – State of the Art and Future Applications," in Proc. 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC), Orlando, FL, 5-7 May, 2008, pp. 436-442.

[3] Elena Lisova, Elisabeth Uhlemann, Wilfried Steiner, Johan Åkerberg, and Mats Björkman, "Risk Evaluation of an ARP Poisoning Attack on Clock Synchronization for Industrial Applications," in Proc. IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan, 14-17 Mar., 2016.

[4] IEEE. (2008). IEEE 1588, "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems". Available: http://www.nist.gov/el/isd/ieee/ieee1588.cfm

[5] Bernd Hirschler and Albert Treytl, "Validation and verification of IEEE 1588 Annex K," in Proc. International IEEE Symposium on Precision Clock Synchroniation for Measurement Control and Communication (ISPCS), Munich, Germany, Sep., 2011.

[6] Markus Ullmann and Matthias Vögeler, "Delay attacks - implication on NTP and PTP time synchronization," in Proc. International Symposium on Clock Synchronization for Measurenment, Control and Communication (ISPCS), Brescia, Italy, Oct., 2009.

[7] Albert Treytl, Georg Gaderer, Bernd Hirschler, and Ron Cohen, "Traps and pitfalls in secure clock synchronization," in Proc. IEEE Internatioal Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS), Vienna, Austria, Oct. , 2007.

[8] Michael Wooldridge, "Does Game Theory Work?," IEEE Intelligent Systems, vol. 27, no. 6, pp. 76-80, Nov.-Dec. 2012.

[9] Sankardas Roy, Charles Ellis, Sajjan Shiva, Dipahkar Dasgupta, Vivek Shandilya, and Qishi Wu, "A Survey of Game Theory as Applied to Network Security," in Proc. 43rd Hawaii international Conference on System Sciences (HICSS), Honolulu, HI 2010, pp. 1-10.

[10] T. Alpcan and T. Basar, "A game theoretic approach to decision and analysis in network intrusion detection," in Proc. 42nd IEEE Conference on Desicion ans Control Hawaii, USA, 9-12 Dec., 2003, pp. 2595-2600.

[11] Sonija Buchegger and Tansu Alpcan, "Security Games for Vehicular Networks," Mobile Compiting, IEEE Transaction on, vol. 10, no. 2, pp. 280-290, 2011.

[12] Tal Mizrahi, "A game theoretic analysis of delay attacks against time synchronization protocols," in Proc. International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communiation (ISPCS), San-Francisco, CA, 24-28 Sep., 2012, pp. 1-6.

[13] Elena Lisova, Marina Gutierrez, Wilfried Steiner, Elisabeth Uhlemann, Johan Akerberg, Radu Dobrin, and Mats Bjorkman, "Protecting Clock Synchronization - Adversary Detection through Network Monitoring," Journal of Electrical and Computer Engineering (JECE), 2016.

[14] S. H. Lin, T. C. Lee, and M. F. Gardina, "Diversity protections for digital radio-summary of ten-year experiments and studies," IEEE Communication Magazine, vol. 26, no. pp. 251-63, 1988.