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

Elena Lisova*, Elisabeth Uhlemann*, Wilfried Steiner‡, Johan Åkerberg*, Mats Björkman*
*Mälardalen University, Västerås, Sweden
‡ TTTech Computertechnik AG, Vienna, Austria
{elena.lisova, elisabeth.uhlemann, johan.akerberg, mats.bjorkman}@mdh.se, wilfried.steiner@tttech.com

***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.**

Keywords— IEEE 1588, synchronization, ARP poisoning

I. 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, 6].

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 II presents the system model and requirements for security solutions targeting the industrial application area. In Section III

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.

we give an overview of the IEEE 1588 standard, its weak spots and its security solutions, whereas a vulnerability analysis is done in Section IV. The ARP poisoning attack is discussed in Section V. In Section VI, the attack analysis, its applicability and consequences are investigated. Section VII presents an overview of possible mitigation techniques for ARP poisoning. An evaluation of the approach and our results are described in Section VIII. Finally, Section IX provides the conclusions and description of future work.

II. 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 [7]. 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 [7], 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.

III. 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 [8]. 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 [8]. 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 [9] 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, 8, 9], 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 [10]. 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 [8] 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.

IV. 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. 1). In reality, however, we cannot guarantee this in any distributed system, and hence it is required that the blue line in Fig. 1, showing the behavior of t_{clock} with respect to t_{real} is within the real time plus, minus Δt_{max}. The clock correction procedure is executed every $\Delta t_{per,}$, and the value of Δt_{per} depends on the clock drifts. Therefore, to be synchronized, the line showing the real dependency (solid green line), is shifted in steps every Δt_{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:

$$\left|t_{clockA} - t_{clockB}\right| > \Delta t_{max} \qquad (1)$$

According to IEEE 1588 [3], a slave clock calculates its offset to the master, Δt_{offM} as:

$$\Delta t_{offM} = t_{inS} - t_{outM} - t_{corr}\,, \qquad (2)$$

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 (1). It is interesting to note that the adversary can only make the slave clock slower, since imposing a delay of the synchronization message implies that t_{inS} is increased, and consequently Δt_{offM} would be increased as well.

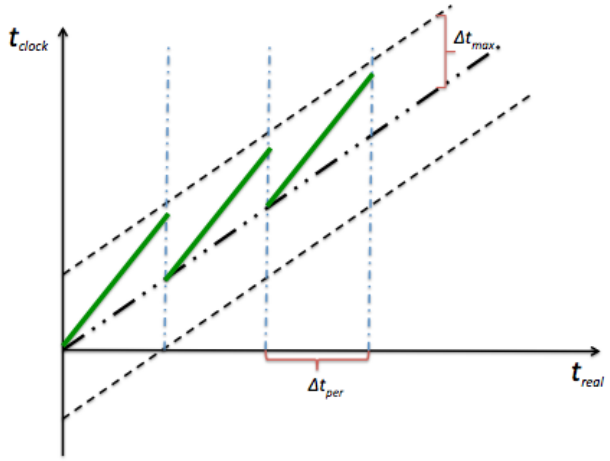

Fig.1 Node clock drift, blue line – the actual time of the clock that should stay within the bounds (dotted lines) in order to keep the node synchronized.

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

$$t_{clock}^{i+1} = t_{clock}^{i} - \Delta t_{offM} \qquad (4)$$

The clock drift is a random value affecting the angle of the green line in Fig 1. Generally, the more expensive a clock is, the lower clock drift is guaranteed by its manufacturer. Let us define slave clock time as:

$$t_{clock} = \alpha \cdot t_{real}\,, \qquad (5)$$

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.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}.

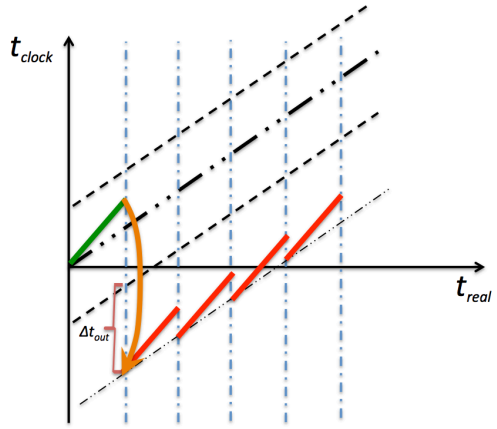

Fig.2 Sustainable delay attack on the faster clock.

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.

V. ARP Poisoning

ARP is a well-known and widely used protocol, and therefore, ARP poisoning is also a commonly used attack method [11]. 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 [12]. 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 [13] 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.

VI. 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 III.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 concrete application and the network architecture. Thus it is reasonable to target a node which 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 networks 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.

VII. 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] [14] several protection techniques were presented. Most existing techniques like [5, 6] and [14] 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] 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 [14]. 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 [9]. 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. This can help to detect an anomaly. 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.

VIII. 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) [15]. 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) [16] 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 Tab. I they are specified as users, whereas IP and MAC addresses of all three are specified as numbers.

TABLE I. 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 (Tab. II).

TABLE II. 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 [17] 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.

IX. 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.

References

[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 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. 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] 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.

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

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

[10] 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.

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

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

[13] 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.

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

[15] 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. CAV*, Edinburgh, Scotland, Jul., 2005.

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

[17] 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.