Game Theory Applied to Secure Clock Synchronization with IEEE 1588

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

***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.**

Keywords—clock synchronization; delay attack; game theory

I. 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 approach 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 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 [11] 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

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.

between an adversary, attacking clock synchronization and a network monitor, proving network protection. Players and their strategies 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 II introduces our system model and explains the idea of Quarantine mode. Next, the game is formulated in Section III, whereas Section IV 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 V. Finally, Section VI concludes the paper.

II. 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 delay, 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 [12]. 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 wireless 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 was found to have an exponential distribution [13]. Hence, we model offsets related to nature with an exponential distribution, since wired 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 [12]. 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 [12]. 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.

III. Proposed Game Model

The main components of a game are actors and their strategies. 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:

$$S = S_{nat} \times S_{adv} \times S_{mon} \ , \tag{1}$$

where $\times$ 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:

$$S_{nat} = \left\{ d_{nat} \,\middle|\, d_{nat} > 0 \right\}, \tag{2}$$

where d_{nat} is a delay caused by 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

$$S_{adv} = \left\{ S_{adv}^{NA}, S_{adv}^{CD}, S_{adv}^{ID}, S_{adv}^{RD} \right\} \ . \tag{3}$$

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.

$$S_{adv}^{NA} = 0 \ . \tag{4}$$

- Constant Delay (CD)

$$S_{adv}^{CD} = \left\{ d_{adv} \,\middle|\, d_{\min} \le d_{adv} \le d_{\max} \right\}, \tag{5}$$

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)

$$S_{adv}^{ID} = \left\{ i \cdot d_{adv} \,\middle|\, 0 < d_{adv} < d_{\max} \right\}, \tag{6}$$

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

- Random Delay (RD)

$$S_{adv}^{RD} = \left\{ (d_{adv,i}, 0) \,\middle|\, 0 < d_{adv,i} < d_{\max} \right\}, \tag{7}$$

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

$$S_{mon} = \left\{ S_{mon}^{AND}, S_{mon}^{Q}, S_{mon}^{AD} \right\} \ . \tag{8}$$

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

$$S_{mon}^{Q} = \left\{ \sigma_{applied} = \sigma_{meas} \cdot \beta \,\middle|\, 0 < \beta < 1 \right\}, \tag{9}$$

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:

$$S_{mon}^{AND} = \left\{ \sigma_{applied} = \sigma_{meas} \right\}. \tag{10}$$

- *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:

$$S_{mon}^{AD} = \left\{ \sigma_{applied} = 0 \right\}. \tag{11}$$

C. 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:

$$p_{NA} + p_{CD} + p_{ID} + p_{RD} = 1 \, , \tag{12}$$

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:

$$p_{NAD} + p_{Q} + p_{AD} = 1 \, , \tag{13}$$

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:

$$\bar{\sigma} = \frac{1}{N} \sum_{i=1}^{N} \sigma_{meas,i} \ , \tag{14}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (\sigma_{meas,i} - \bar{\sigma})^2}{N}} \ , \tag{15}$$

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. 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 $\overline{\Sigma}_L$ and $\overline{\Sigma}_H$ are the low and high thresholds for the mean respectively. If one of the indicators is above the corresponding low threshold but 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: $\overline{\sigma} > \overline{\Sigma}_L$ 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: $\overline{\sigma} > \overline{\Sigma}_L$ 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: $\overline{\sigma} > \overline{\Sigma}_H$ 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.

IV. Analys 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:

$$f_{ChD}(d_{ChD}) = \lambda \cdot e^{-\lambda d_{ChD}},\ d_{ChD} \geq 0, \tag{16}$$

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 (CD) 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:

$$D_{CD} = \left\{ d_{CD,j} \right\}_{j=1}^{N_1}, \tag{17}$$

$$\sum_{j=1}^{N_1} p_{CD,j} = 1,\ p_{CD,j} = f_{CD}(d_{CD,j}), \tag{18}$$

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:

$$f_{offset}(\sigma_{meas}) = \sum_{j=1}^{N_1} f_{ChD}(\sigma_{meas} - d_{CD,j}) f_{CD}(d_{CD,j}) = \\ = \lambda e^{-\lambda \sigma_{meas}} \sum_{j=1}^{N_1} (e^{\lambda d_{CD,j}} \cdot p_{CD,j}). \tag{19}$$

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

$$k_{CD} = \sum_{j=1}^{N_1} (e^{\lambda d_{CD,j}} \cdot p_{CD,j}), \tag{20}$$

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

$$f_{offset}(\sigma_{meas}) = k_{CD} \lambda e^{-\lambda \sigma_{meas}}. \tag{21}$$

B. ID strategy

In the next case, we consider a linearly increasing delay (ID) 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:

$$D_{ID} = \left\{ i \cdot d_{ID,j} \right\}_{j=1}^{N_2}, \tag{22}$$

$$\sum_{j=1}^{N_2} p_{ID,j} = 1,\ p_{ID,j} = f_{ID}(d_{ID,j}), \tag{23}$$

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:

$$f_{offset}(\sigma_{meas}) = k_{ID} \lambda e^{-\lambda \sigma_{meas}}, \tag{24}$$

$$k_{ID} = \sum_{j=1}^{N_2} (e^{\lambda \cdot i \cdot d_{ID,j}} \cdot p_{ID,j}). \tag{25}$$

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:

$$f_{RD}(d_{RD}) = \frac{1}{d_{\max}},\ d_{RD} \in (0, d_{\max}]. \tag{26}$$

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

$$f_{offset}(\sigma_{meas}) = \int_{-\infty}^{+\infty} f_{RD}(\sigma_{meas} - \tau) f_{ChD}(\tau) d\tau = \\ = \frac{\lambda e^{-\lambda \sigma_{meas}}}{d_{\max}} \int_0^{d_{\max}} e^{\lambda \tau} d\tau = \frac{e^{-\lambda \sigma_{meas}}}{d_{\max}} (e^{\lambda d_{\max}} - 1) \tag{27}$$

.

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

$$f_{offset}(\sigma_{meas}) = k_{RD}\lambda e^{-\lambda\sigma_{meas}}, \quad (28)$$

$$k_{RD} = \frac{1}{\lambda d_{\max}}(e^{\lambda d_{\max}} - 1). \quad (29)$$

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 γ percent. 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 γ percent 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:

$$\Sigma_L = \frac{k_{ID}}{\lambda^2}, \ \Sigma_H = \frac{k_{ID}}{\lambda^2}(1+\gamma), \quad (30)$$

$$\overline{\Sigma}_L = \frac{k_{ID}}{\lambda}(1+\gamma), \ \overline{\Sigma}_H = \frac{k_{ID}}{\lambda}(1+2\gamma). \quad (31)$$

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.

V. 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.

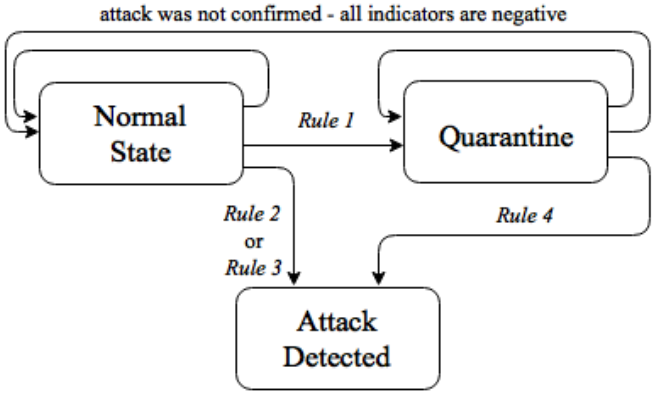

Fig. 1 States of the monitor and transitions between them.

Fig. 1 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. 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 (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.

To compare the three types of considered delay attacks, we need to compare the values of the related detection coefficient from equations (20, 25 and 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. 2 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. 2 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. 2. 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. 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. 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.

VI. 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 configuration. 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.

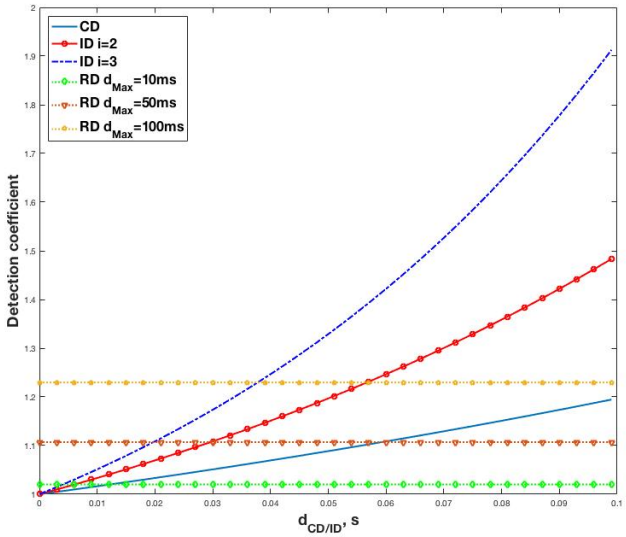

Fig. 2 Detection coefficients for RD, ID, and RD types of attacks.

References

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

[2] W. Elmenreich, "Time-Triggered Fieldbus Networks – State of the Art and Future Applications," *in Proc. ISORC*, Orlando, FL, 5-7 May, 2008.

[3] 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. ICIT*, Taipei, Taiwan, 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] B. Hirschler and A. Treytl, "Validation and verification of IEEE 1588 Annex K," *in Proc. ISPCS*, Munich, Germany, Sep., 2011.

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

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

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

[9] 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 HICSS*, Honolulu, HI 2010.

[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.

[11] T. Mizrahi, "A game theoretic analysis of delay attacks against time synchronization protocols," *in Proc. ISPCS*, San-Francisco, Sep., 2012.

[12] E. Lisova, M. Gutierrez, W. Steiner, E. Uhlemann, J. Akerberg, R. Dobrin, and M. Bjorkman, "Protecting Clock Synchronization - Adversary Detection through Network Monitoring," *JECE,* 2016.

[13] 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, 1988.