Technical Report TR-SUT-CE-23-01, May 2023

"New Calculations Along with New Routes to Improve TDA Detection in PTP Network"

Mohsen Moradi[1], *Amir Hossein Jahangir*[2]

Department of Computer Engineering,
Sharif University of Technology
[1] *fmoradi@ce.sharif.ir* , [2] *jahangir@sharif.ir*

Abstract: When using Precision Time Protocol, PTP, as one of the most accurate and well-known protocols for clock synchronization, Time-Delay Attacks, TDAs, are a challenging concern on the way. In order to result in a robust synchronisation through PTP, detecting different types of TDAs is a great favour. As adding new paths is recommended in valuable research works, this report considers adding new paths to the network and studies the result on TDA detection. Then to reach to a more robust algorithm, new path delay calculations is also considered. The report explores how these two wings result is a sustained rise towards the security peak.

Keywords — Synchronization; Precision Time Protocol; Time-Delay Attack; Attack Detection.

1. Introduction

1.1. Aim and area

Precision Time Protocol (PTP) is an accurate and straight path that leads us to the desired synchronization destination. Cyber-attacks play the role of potholes which may disrupt synchronization of the network through different strategies such as data manipulation, spoofing, replay attack, interception and information removal, message delay manipulation, forgery of network clocks, and Denial-of-Service (DoS) [1, 2].

Applying different types of Time-Delay Attacks (TDAs) is a common craft by the network attackers to confuse PTP. Among so many research works which have tried to improve TDA detection methods in PTP, the methods in [1] and [2] are two of the most effective ones. This technical report is investigating these two methods to take a step toward superior TDA detection algorithm. With this goal, the main driver of the report is investigating more types of TDAs, specially the attacks on Report message which was neglected.

The report continues as follows. Firstly, the research perspective presents PTP introduction and different methods to result in a more robust PTP. Section 2 investigates a security challenge in the previous reporting-based methods. Section 3 explores new strategies to improve detection ability and studies special TDAs which previous method fail to detect. The detection procedure proposed by this report is presented in this section. The simulation results are presented and discussed in section 4. Finally, the report is concluded in section 5.

1.2. Research perspective

Precision Time Protocol, PTP, was firstly introduces in 1990s which was later presented as IEEE 1588 standard at 2002. The interested reader may refer to [3] to [7] for more information about PTP and IEEE 1588 standard as well as basic concept and math.

As PTP was opening its way to grow as a well-known synchronization protocol in the distributed and intelligent networks, so did the security concerns. Tsang and Beznosov [8, 9] introduced PTP security through averaging the measured delays. Later, security solutions were presented along with the PTP in different aspects. Various research works studied TDAs and PTP-based network resilience and robustness against it. As categorized in [2], PTP-based network security solutions to detect or encounter TDAs may be of the following categories:

1- Monitoring sudden changes in the delay value or the offset obtained by the network clocks [8, 9];
2- Round Trip Delay (RTD) measurement [10, 11, 12];
3- Message encryption and adding cover traffic [13, 14, 15];
4- Using multiple Grand Master Clocks (GMCs) and multiple paths to send the messages [16, 17];
5- Network clocks offset comparison in the presence of an external reference clock [1];
6- Network clocks offset comparison with each other [2];
7- Other researches.

The subject under investigation in this report is related to the reports from the network clocks. Report-based methods cover nonstop monitoring of synchronization status. This is so important in susceptible systems which is a great superiority of over other TDA detection methods of other categories. Further, all attack types having different specifications may be detected with the help of report-based methods [1]. Therefore, special practical capabilities of report-based methods attract a lot of attention and this report is studying debilities of the previous robust methods to look for more reliable and robust methods in this regard. With this goal in the mind, we start with investigating the report-based methods presented in [1] and [2].

Reporting from network clocks and their communication builds the basis of the methods in these references.

The report-based methods check the offset of the Slave Clocks (SCs) with respect to the Master Clock (MC). TDA detection through offset-check based on the network clocks' reports in PTP networks was introduced in [1]. To do so, a new GMC-synchronized foolproof reference node was added to the network to let check the SCs offset, called "*Network Time Reference (NTR)*". As an impractical assumption, the new node was supposed to be foolproof. On the other side, reference [2], applied checking the offsets of SCs relative to MC. It considered a timeless pure-analyst network unit called "*Analaysis Unit (AU)*" without the need for security-threatening NTR.

2. Security hole in existing report-based algorithms

These methods presented in [1] and [2] are two of the best methods proposed in this field, as far as this report has researched. These methods have pros and cons over each other, but they share a similar problem, "neglecting the attacks on the report message". In other words, the report message is assumed to be foolproof. This research concentrates on the report message and different TDAs applied on it. The method and security concerns of [1] and [2] may be concluded as Table 1.

As Table 1 shows, the method in [1] uses network clustering and also considers an accurate NTR. This method neglects TDAs on the report message, considers accurate NTD, fails to detect simultaneous TDAs, and do not consider the location of the attack. The other work in [2] uses network block building and considers timeless AU. This method also neglects TDAs on the report message, but thanks to the block building, it is able to determine the under-attack block (better than no-location) as well as simultaneous TDAs (with an independent attack detection operation).

The research work in [2] tried to solve some of the security issues of the method in [1] through a new method and this report is investigating improvements that may resolve any security concerns of the method in [2]. Therefore, the need for further investigation of the method in [2] is sensed. In the following, more details about reference [2] is presented.

The related definitions and concepts used or introduced in [2] are as follows:

- *BMC (Best Master Clock)*: PTP Synchronization process first applies BMC algorithm to create a hierarchical Master-Slave structure in the network.
- *GMC (Grand Master Clock)*: This clock a uses time reference appliance (such as GPS) as synchronization while the GMC itself is a synchronization source for other network clocks.
- *Master and Slave ports*: Master node supply time information for the other nodes connected to them, known as Slave ports.
- *Passive port*: A port that is not in Master nor Slave state.
- *OC (Ordinary Clock)*: A clock connected to the end device playing GMC or Slave role. OC cannot be a

Table 1 Methods and security holes of [1] and [2]

	Reference [1]	*Reference [2]*
Applied methods	- Adding NTR - Clustering	- Network block building - Considering AU
Security concerns	- TDA on report message - Foolproof NTR - Non-simultaneous detection - No-location detection	- TDA on report message - Approximate (block-level) location detection

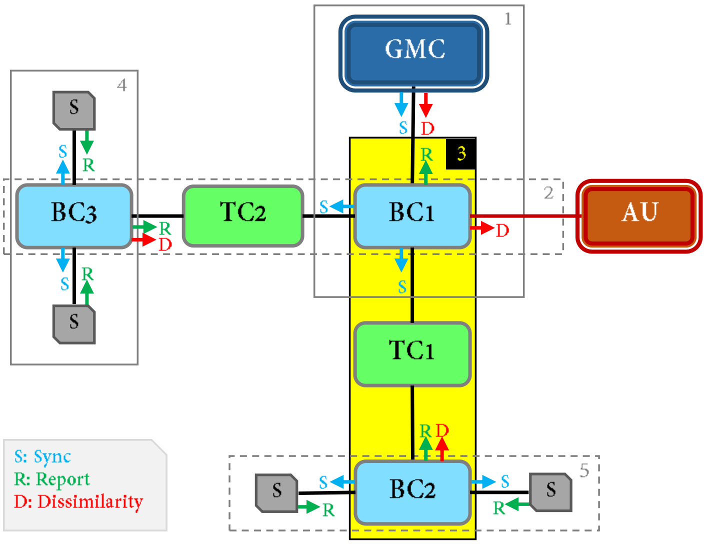

Fig. 1. Network block building procedure in [2]

network switch and only sends/receives IEEE1588 messages.

- *BC (Boundary Clock)*: A network switch which is a Slave/Master for its Masters/Slaves. BC may have multiple Master ports, but only one Slave port.
- *Residence time*: The time duration from entrance to the exit of an IEEE 1588 message in a network clock. SCs are updated using residence time.
- *TC (Transparent Clock)*: A network switch with the ability to measure the message residence time and set it as the *Correction Field* (CF) of the IEEE 1588 messages before sending them (non-IEEE 1588 messages are sent in a normal switch mode). TC ports are not Master or Slave in the Master-Slave structure.
- *Network blocks*: smaller parts of the network in which it is divided. Each block is defined to have at most two BCs and one (or no) TC.
- *HC (Head Clock) and RC (Root Clock)*: Considering the BMC algorithm, one of the BCs in each block would be the Master and the other one the Slave. The Master is called HC, and the Slave, RC.
- *Dissimilarity message*: The message consisting the Calculated Offset (CO) measured by HC of each block as the difference between HC and its corresponding Slave(s).

The network block building and messages are presented in Fig. 1. Based on the network block building, the synchronization and TDA detection process in [2] may be stated in the following steps:

1) Applying the network block building;
2) Considering BMC to determine the roles for nodes and ports;
3) Sending report messages for RC to the HC in each block;
4) Calculating CO by the HC of each block and sending it as dissimilarity message to be analyzed in AU.
5) Simultaneously detection of the under-attack blocks in AU. An offset of $nv \approx 0$ (negligible value - an extremely small value approximating zero) means no-attacks (white status) while non-zero offsets confirm attack influence (red status). TDAs, attacks on TC, and message residence time changes are all shown to be detectable.

3. Exploring new ways

Considering related sources in the network synchronization field brings some insights to the mind which may be the route towards more robust algorithms. Developing new paths and assigning new roles to collect more date enabling more calculations seem to be the ways worth exploring. It is predicted to reach a more robust algorithm at a cost of an overhead traffic. Therefore, in the following sections, modifying the method in [2] is investigated in the hope that a more robust algorithm will be resulted with negligible overhead traffic.

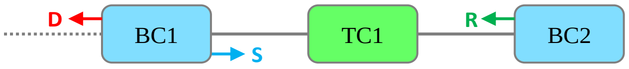

Fig. 2. Messages in the block under investigation

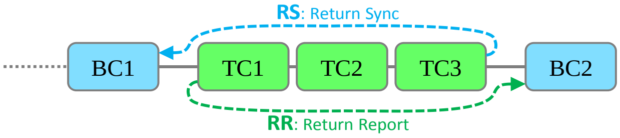

Fig. 3. New paths in the block under investigation

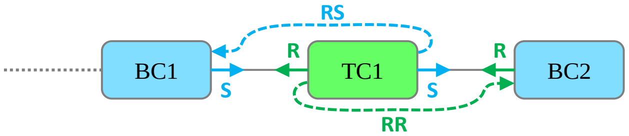

Fig. 4. Sync and Report messages paths

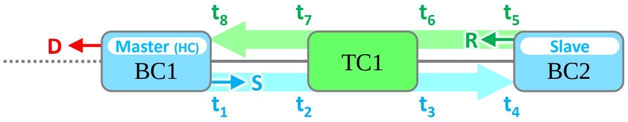

Fig. 5. New considered timestamps and tasks

3.1. Developing new paths in the network

IEEE Standard 1588-2019 recommends new paths to result in robust synchronization through sending the Sync message from the Mater to Slave [4]. As a new adaptation of the idea, this report considers new efficiently added paths to help with TDA detection procedure, but here the added paths do nothing about the network synchronization.

Consider the sample block of Fig. 2 (simple rotation of block number 3 in Fig. 1) to start looking over new paths. Now, consider the new paths shown as dashed line in Fig. 3 as follows:

1) *RR (Return Report)*: from RC to HC-adjacent TC (i.e., the TC close to HC);

2) *RS (Return Sync)*: from HC to RC-adjacent TC (i.e., the TC close to RC);

So, Sync and Report messages pass through the network as shown in Fig. 4. Note that these new paths differ from the recommended paths in PTP version 2.1.

3.2. New path delay calculations

As mentioned, gathering more data through assigning new roles to the network nodes could result in more traces of TDAs which open the way to detect the attacks with applying new purposeful calculations.

Now, consider a TDA on the Sync message along with an attack on TC (report message residence time change). To do so, suppose that the attacker has sent a Sync with a delay of μ (a positive value) and then increased the report message residance time exactly by the same value of μ to keep changes secret at the final computation and analysis. In this scenario, if the attack on the Sycn is applied after TC, the attacker succeded to “clear TDA traces” and AU will notice to TDAs. Also, with the same attack stategy, suppose that the attacker has increased the Sync message residence time by μ (i.e., attack on TC) and sent the Report message with the same value of delay (TDA on Report). So, if TDA on the Report is applied after the Report passes through TC, nothinh will be mentioned in AU.

As mentioned, to shed more light on the network paths and cath TDA red-handed, more data is needed. So, as shown in Fig. 5, some timestamps in the network and some new tasks are needed as follow to enable new calculations:

- *BC1 (HC)→Master*

  - registering all timestamps of Fig. 5 in dissimilarity;
  - sending dissimilarity to AU.

- *Master-adjacent TC: TC1*
  - registering Sync entrance time (t_2) and Report exit time (t_7) in the Report message.

- *Slave-adjacent TC: again TC1*
  - registering Sync exit time (t_3) and Report entrance time (t_6) in the report message.

- *Slave: BC2*
  - registering Sync entrance time (t_4), Report exit time (t_5), and CO, in the report message.

As intended, Report message has got CO and timestamps [t_2 , t_3, … , t_7] to deliver to HC (here, BC1) while HC is aware of [t_1 , t_8]. Therefore, HC registers all eight timestamps in the dissimilarity message to be handed to AU. In AU new calculations will take part using the new timestamps. The authors intend path delay calculation usind two different ways to let compare hidden delay changes. So, P2P (Point-to-Point) and E2E (End-to-End) methods are considered to compare the results.

The path delay calculations through the E2E method is done in AU based on the following relations:

$$\begin{cases} \lambda_{BC1-TC1} = \frac{1}{2}\Big((t_2 - t_1) + (t_8 - t_7)\Big) \\ \lambda_{TC1-BC2} = \frac{1}{2}\Big((t_4 - t_3) + \big(t_6 - (t_5 + \delta)\big)\Big) \end{cases} \tag{1}$$

In the relations, Report sending time is modified using the time before the update by Slave (δ is added to t_5). This is because the values of the timestamps t_4 and t_5 are to have the same time scale, while Slave clock updates its time after receiving Sync.

Now, with E2E calculations in hand, the results must be equal to that of P2P method.

3.3. TDA detection procedure

In the previous sections, the proposed ideas toward a more capable TDA detection algorithm were introduced. Two main ideas are concluded as: adding new paths, and applying new calculations. Each of these ideas cover the ability to detect more TDA types. The following sections will examine how each of these ideas help detecting TDAs.

3.3.1. Assistance of the return paths

Here, the sole efficiency of the new added paths is examined and E2E calculations will not participate.

TC sends messages from all of its ports after updating CF. So, the messages sent from TC have the same characteristics. Sync and Report messages are sent back to their source, so send and receive time is a clue to detect TDAs through offset measurement.

Suppose that there is really no TDA. So, Sync and Report flow through the network as previously was shown in Fig. 4. Now, consider that following about Sync:

1) BC1 sends Sync @ $t_{BC1} = t_{GMC}$

2) RS is back to BC1 @ $t^{+}_{BC1} = t_{GMC} + \lambda_{BC1-TC1} + \rho_1 + \lambda^{+}_{TC1-BC1}$

3) Message CF when reaching BC1 is $CF^{+}_{Sync} = \lambda_{BC1-TC1} + \rho_1$

Where @ is used to show the time, RS is the Return Sync, the plus superscript ($^{+}$) means the new added paths are considered in relation, and ρ_1 is the message residence time. So, offset will be measured in BC1 as follows:

$$\begin{aligned}
&\delta^{+}_{BC1} = t^{+}_{BC1} - t_{BC1} - (CF^{+}_{Sync} + \lambda^{+}_{TC1-BC1}) \\
&\Rightarrow \delta^{+}_{BC1} = t_{GMC} + \lambda_{BC1-TC1} + \rho_1 + \lambda^{+}_{TC1-BC1} \\
&\qquad - (t_{GMC} + \lambda_{BC1-TC1} + \rho_1 + \lambda^{+}_{TC1-BC1}) \\
&\Rightarrow \delta^{+}_{BC1} = nv \approx 0
\end{aligned} \tag{2}$$

Where the offset is resulted to be extremely small (shown with *nv*; negligible value) as no TDAs had been applied (note that in real situation the result will be a minute value approaching zero).

Applying the same calculations about Report message will be as follows:

1) BC2 sends Report @ $t_{BC2} = t_{GMC}$

2) RR is back to BC2 @ $t^{+}_{BC2} = t_{GMC} + \lambda_{BC2-TC1} + \rho_1 + \lambda^{+}_{TC1-BC2}$

3) Message CF when reaching BC2 is $CF^{+}_{Sync} = \lambda_{BC2-TC1} + \rho_1$

Where RR is Return Report. Then offset in BC2 will be:

$$\begin{aligned}
&\delta^{+}_{BC2} = t^{+}_{BC2} - t_{BC2} - (CF^{+}_{Report} + \lambda^{+}_{TC1-BC2}) \\
&\Rightarrow \delta^{+}_{BC2} = t_{GMC} + \lambda_{BC2-TC1} + \rho_1 + \lambda^{+}_{TC1-BC2} \\
&\qquad - (t_{GMC} + \lambda_{BC2-TC1} + \rho_1 + \lambda^{+}_{TC1-BC2}) \\
&\Rightarrow \delta^{+}_{BC2} = nv \approx 0
\end{aligned} \tag{3}$$

Where the same result is given due to white status. As the calculated offset during white status was really small, it is time to check the results when different types of attacks have been applied as presented in the following.

Consider TDA on Sync message before passing TC1 and also an attack on TC where the attacker sends Sync with a delay of μ , and to prevent detection, increases Report residence time by μ too. The offset calculation in BC2 will result as:

$$\begin{aligned}
&\delta^{+}_{BC2} = t^{+}_{BC2} - t_{BC2} - (CF^{+}_{Report} + \lambda^{+}_{TC1-BC2}) \\
&\Rightarrow \delta^{+}_{BC2} = t_{GMC} - \mu + \lambda_{BC2-TC1} + \rho_1 + \lambda^{+}_{TC1-BC2} \\
&\qquad - \mu - (t_{GMC} + \lambda_{BC2-TC1} + \rho_1 + \lambda^{+}_{TC1-BC2}) \\
&\Rightarrow \delta^{+}_{BC2} = -\mu
\end{aligned} \tag{4}$$

Where TDA may be detected due to non-zero offset. In this type of attack, if the attacker applies a delay if μ on RR before reaching BC2, the offset will be calculated as:

$$\begin{aligned}
&\delta^{+}_{BC2} = t^{+}_{BC2} - t_{BC2} - (CF^{+}_{Report} + \lambda^{+}_{TC1-BC2}) \\
&\Rightarrow \delta^{+}_{BC2} = t_{GMC} - \mu + \lambda_{BC2-TC1} + \rho_1 + \lambda^{+}_{TC1-BC2} + \mu \\
&\qquad - \mu - (t_{GMC} + \lambda_{BC2-TC1} + \rho_1 + \lambda^{+}_{TC1-BC2}) + \mu \\
&\Rightarrow \delta^{+}_{BC2} = nv \approx 0
\end{aligned} \tag{5}$$

So, a smart attack can result in a hidden red status as the offset is mistakenly calculated to approach to zero. Now, let's check the offset for RS. Here, BC1 measures the offset as:

$$\begin{aligned}
&\delta^{+}_{BC1} = t^{+}_{BC1} - t_{BC1} - (CF^{+}_{Sync} + \lambda^{+}_{TC1-BC1}) \\
&\Rightarrow \delta^{+}_{BC2} = t_{GMC} + \lambda_{BC1-TC1} + \mu + \rho_1 + \lambda^{+}_{TC1-BC1} \\
&\qquad \mu - (t_{GMC} + \lambda_{BC1-TC1} + \rho_1 + \lambda^{+}_{TC1-BC1}) \\
&\Rightarrow \delta^{+}_{BC2} = \mu
\end{aligned} \tag{6}$$

Here, if the attacker wants to hide TDA, a delay of $-\mu$ must be applied on RS which means sending it sooner ! As this is impossible, TDA will be detected in this case.

What if the attacker attack Sync after passing TC1 ? Or what about the other possible attach types and locations? Investigation of all types and locations of the attacks is out of the space limit of this report, but examining an example as mentioned above confirmed the hidden red status and the vital need to apply other methods to detect TDAs. Therefore, the following section considers the other wing of the new algorithm considered in this report which is E2E calculation.

3.3.2. Assistance of return paths along with new calculations

Investigating two special timestamps t_4 (the last timestamp for Sync on its path) and t_8 (the last timestamp for Report on its path) which checks TDAs after passing through TC, will hand in new information.

To take a better look, consider that a TDA as large as μ is applied on Report after TC1 in Fig. 5. So, Report is effected with a delay of μ when coming to BC1. The path delay with using E2E method will be as follows:

$$\begin{aligned}\lambda'_{BC1-TC1} &= \tfrac{1}{2}\Big((t_2 - t_1) + \big((t_8+\mu) - t_7\big)\Big)\\ &= \tfrac{1}{2}\Big((t_2 - t_1) + (t_8 + t_7)\Big) + \tfrac{1}{2}\mu \\ &= \lambda_{BC1-TC1} + \tfrac{1}{2}\mu\end{aligned} \tag{7}$$

Where λ' means λ_{E2E}. So, for this scenario (with TDA of μ on Report after TC1) it will be concluded that:

$$\begin{cases}\text{white status}\\ \text{red status}\end{cases} \Rightarrow \begin{cases}\lambda_{\mathrm{E2E}} = \lambda_{\mathrm{P2P}}\\ \lambda_{\mathrm{E2E}} = \lambda_{\mathrm{P2P}} + \tfrac{1}{2}\mu\end{cases} \tag{8}$$

The abovementioned conclusion is also true for the scenario in which TDA is applied on Sync after TC1. In this case, $\lambda_{\mathrm{E2E}} = \lambda_{\mathrm{P2P}} + \frac{1}{2}\mu$ between RC and the nearest TC (TC1).

As stated, to reach to a conclusion, consider the following TDAs and the result:

$$\begin{cases}\text{Sync attack after TC1}\\ \text{Report attack after TC1}\end{cases} \Rightarrow \begin{cases}\text{change of } t_4\\ \text{change of } t_8\end{cases} \Rightarrow \lambda_{\mathrm{E2E}} \neq \lambda_{\mathrm{P2P}} \tag{9}$$

So, comparison of the calculated delay with using E2E and P2P will be a helpful method to detect the hidden TDAs when applying only new paths. The two wings of the robust TDA detection method are now completely described. In the following section, TDA detection procedure of the new robust method will be presented.

3.4. TDA detection procedure in the new robust method

The proposed TDA detection procedure as the robust detection algorithm is shown in Fig. 6. As seen, after BMC is applied and AU is informed of the structure, structure revision is checked by AU to reconsider block building if revised. After messages flow and data feedback, AU

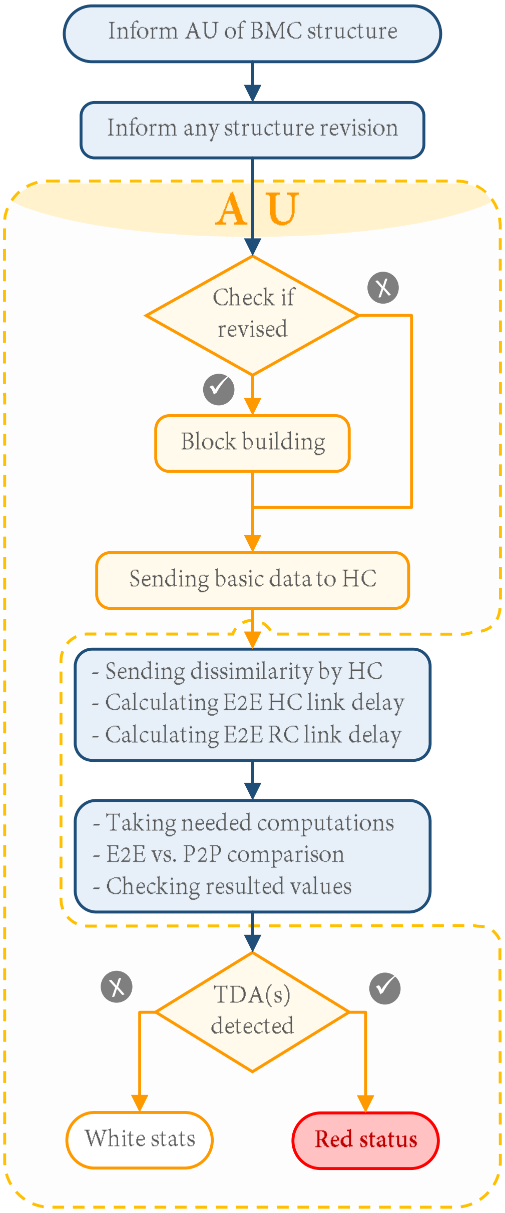
Inform AU of BMC structure
Inform any structure revision
A U
Check if revised
Block building
Sending basic data to HC
- Sending dissimilarity by HC
- Calculating E2E HC link delay
- Calculating E2E RC link delay
- Taking needed computations
- E2E vs. P2P comparison
- Checking resulted values
TDA(s) detected
White stats
Red status

Fig. 6. Proposed TDA detection procedure

searches for any abnormal calculated delay in the defined parameters and if the delay value is not in the order of $nv \approx 0$ (i.e., a predefined extremely small value), TDA(s) will be detected and red status announced.

3.4.1. Attack scenarios

Considering the sample network in Fig. 7, the attack scenarios are outlined in Table 2. As shown in Table 2 , there are 18 possible attack scenarios. In the first 6 scenarios, a single attack has been applied without further attempt to hide it. In scenarios 7 to 12, the attacker attempts to hide the first attack by a second attack, and in scenarios 13 to 18, the attacker still hopes to hide the TDAs. The letter "R" in parentheses in the third attack shows that the attack has occurred on the "Return" path. As a shortcoming, the attacks studied in [1] and [2] only include the first 6 scenarios, and further attempts of the attacker are missed.

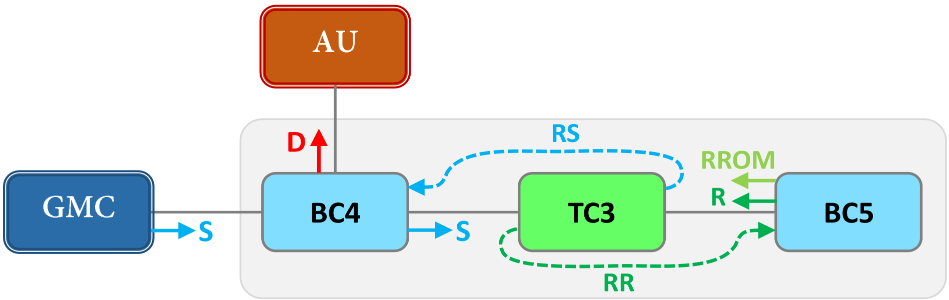

Fig. 7. Graph of PTP-based sample network and new paths to return messages to their source

Table 2. Attack scenarios

Attack scenario	1st Attack		2nd Attack		3rd Attack	
	link or switch	type	link or switch	type	link or switch	type
1	BC4-TC3	Sync	–	–	–	–
2	TC3	Sync	–	–	–	–
3	TC3-BC5	Sync	–	–	–	–
4	BC5-TC3	Report	–	–	–	–
5	TC3	Report	–	–	–	–
6	TC3-BC4	Report	–	–	–	–
7	BC4-TC3	Sync	TC3	Report	–	–
8	TC3-BC5	Sync	TC3	Report	–	–
9	TC3	Sync	BC5-TC3	Report	–	–
10	TC3	Sync	TC3-BC4	Report	–	–
11	TC3	Sync	TC3	Report	–	–
12	TC3	Sync	TC3	Report	–	–
13	TC3	Sync	TC3	Report	TC3-BC5 (R)	Report
14	TC3	Sync	TC3	Report	TC3-BC4 (R)	Sync
15	BC4-TC3	Sync	TC3	Report	TC3-BC5 (R)	Report
16	TC3-BC5	Sync	TC3	Report	TC3-BC5 (R)	Report
17	TC3	Sync	BC5-TC3	Report	TC3-BC4 (R)	Sync
18	TC3	Sync	TC3-BC4	Report	TC3-BC4 (R)	Sync

4. Simulation Results

The result of the attack scenarios is presented in Table 3 (all values are in milliseconds and $nv \approx 0$). The parameters used in Table 3 are defined in Table 4. As can be seen in Table 3, if detection relies only on CO and considers the non-zero value of CO as the attack detection criterion, the attacker can force the value of CO to zero by applying a second attack on the network. Therefore, relying solely on CO value is insufficient for detecting an attack. To effectively address this concern, all arrays within the vector [MLD, SLD, RSO,

Table 3. The outcome of the attack scenarios (all values in milliseconds)

# Scenario	MLD		SLD		MLD	SLD	RSO	RRO	CO	Success		
	P2P	E2E	P2P	E2E	Difference	Difference				[1]	[2]	Proposed
1	2.3	3.3	2.9	2.9	1	*nv*	2	*nv*	2	✓	✓	✓
2	2	2	1.6	1.6	*nv*	*nv*	-2	*nv*	-2	✓	✓	✓
3	4.6	4.6	2.2	3.2	*nv*	1	*nv*	*nv*	2	✓	✓	✓
4	3.2	3.2	3.4	4.4	*nv*	1	*nv*	2	2	✓	✓	✓
5	2	2	2.3	2.3	*nv*	*nv*	*nv*	-2	-2	✓	✓	✓
6	4.4	5.4	3.3	3.3	1	*nv*	*nv*	*nv*	2	✓	✓	✓
7	1.8	2.8	4	4	1	*nv*	2	-2	*nv*	×	×	✓
8	3.3	3.3	1.7	2.7	*nv*	1	*nv*	-2	*nv*	×	×	✓
9	2.5	2.5	4.5	5.5	*nv*	1	-2	2	*nv*	×	×	✓
10	2.4	3.4	4.3	4.3	1	*nv*	-2	*nv*	*nv*	×	×	✓
11	3.9	3.9	2.9	2.9	*nv*	*nv*	2	-2	*nv*	×	×	✓
12	2.6	2.6	4.9	4.9	*nv*	*nv*	-2	2	*nv*	×	×	✓
13	2.9	2.9	2.2	2	*nv*	*nv*	2	*nv*	*nv*	×	×	✓
14	3.8	3.8	4.2	4.2	*nv*	*nv*	*nv*	2	*nv*	×	×	✓
15	1	2	2.3	2.3	1	*nv*	2	*nv*	*nv*	×	×	✓
16	4	4	1.2	2.2	*nv*	1	*nv*	*nv*	*nv*	×	×	✓
17	2.1	2.1	2.2	3.2	*nv*	1	*nv*	2	*nv*	×	×	✓
18	3	4	3.5	3.5	1	*nv*	*nv*	*nv*	*nv*	×	×	✓

Table 4. Effective parameters in the attack detection

Parameter	*Description*
MLD (P2P)	Link delay between BC4 and TC3 obtained through the P2P method
MLD (E2E)	Link delay between BC4 and TC3 was obtained using t_1 to t_8 times stored in the Sync and Report packets
SLD (P2P)	Link delay between TC3 and BC5 obtained through the P2P method
SLD (E2E)	Link delay between TC3 and BC5 was obtained using t_1 to t_8 times stored in the Sync and Report packets
MLD Difference	Difference between MLD (P2P) and MLD (E2E)
SLD Difference	Difference between SLD (P2P) and SLD (E2E)
RSO	Offset calculated by R-Sync
RRO	Offset calculated by R-Report
CO	Offset calculated by BC4 after receiving the Report packet from BC5

RRO, CO] need to be assessed. If any of these values are non-zero, then a network attack can be confirmed. Further, upon comparison of the results, it can be seen that each scenario yields a distinct results matrix. As such, analyzing this matrix can also pinpoint the exact location of the attacks. In the initial six scenarios, only one attack is carried out by the attacker on the sample network. So, in all of these scenarios, CO value is non-zero as the attacker did not run another attack to conceal the initial attack. Therefore, along with the non-zero CO value in these scenarios, MLD and RSO values in the first scenario, RSO value in the second scenario, SLD value in the third scenario, SLD and RRO values in the fourth scenario, RRO value in the fifth scenario and MLD value in the sixth scenario also remain non-zero. Consequently, in these six scenarios, the proposed method can effectively detect the attack.

In scenarios 7 to 12, the attacker launches a second attack to hide the first attack, which causes the value of CO in BC4 to be *nv* (almost zero). Although the value of CO is equal to *nv* in these scenarios, based on the proposed algorithm, the presence of an attack is signaled by any non-zero value in the other parameters MLD, SLD, RRO, and RSO. As shown in Table 3, in scenario 7, the values of MLD, RSO, and RRO; in scenario 8, the values of SLD and RRO; in scenario 9, the values of SLD, RSO, and RRO; in scenario 10, the values of MLD and RSO; and in scenarios 11 and 12 the values of RSO and RRO remain non-zero. So, the algorithm has been able to successfully detect the attack. Finally, in scenarios 13 to 18, the attacker launches the third attack in an effort to hide the prior two attacks. Notably, the count of non-zero parameters in these scenarios has decreased. However, there is still at least one non-zero parameter in each scenario, so the attacker is not successful to hide the attacks in any of the scenarios.

5. Conclusion

As an accurate and widely-used synchronization protocol, PTP has alwaye suffered from malicious network attacks one of which is the well-known TDA. In order to detect TDAs, so many research works and efforts were taken, but a missing part in all of them was TDAs on the Report message.

In this technical report, two main strategies have been considered and explored to reinforce previouse methods. The first strategy designes some paths with a new application in the network, and the secound, applies E2E calculations based on the new data gathered due to the new network tasks assigned to differet nodes. While the first strategy partially improved detection ability, the secound stategy applied along with the first one, resulted in a complete success of the method in detectind different TDAs.

References

[1] M. Bassam, K. Marthe, H. Rachid, D. Mourad and A. Chadi, “An Extension to the Precision Time Protocol (PTP) to Enable the Detection of Cyber Attacks,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 18-27, 2019.

[2] M. Moradi and A. H. Jahangir, “A New Delay Aattack Detection Algorithm for PTP Network in Power Substation,” *International Journal of Electrical Power & Energy Systems*, vol. 133, 2021, 107226.

[3] IEC, “IEC (International Electrotechnical Commission) 61588, Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” IEEE International Standard, Institute of Electrical and Electronics Engineers, 2004.

[4] “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE Standard 1588-2019 (Revision of IEEE Standard 1588-2008),” Institute of Electrical and Electronics Engineers, Inc., IEEE Instrumentation and Measurement Society, New York, USA, 2020.

[5] I. TC57, “IEC 61850: Communication Networks and Systems for Power Utility Automation,” International Electrotechnical Commission Standard, 2010.

[6] U. Keten, "GPS/GNSS Independent Time Transfer Over Telco IP Core Networks Using DTM Overlay," IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS) |, Nov. 2021.

[7] G. M. Garner and H.(E.) Ryu, "Synchronization of Audio/Video Bridging Networks Using IEEE 802.1AS," Samsung Advanced Institute of Technology, Synchronization Over Ethernet and IP Networks, IEEE Communications Magazine, Feb. 2011.

[8] J. Tsang and K. Beznosov, “A Security Analysis of the Precise Time Protocol (Short Paper),” International Conference on Information and Communications Security, 2006.

[9] K. Beznosov and J. Tsang, “Technical Report LERSSE-TR. Security Analysis of the Precise Time Protocol,” University of British Columbia, Vancouver, Canada, 2006.

[10] M. Ullmann and M. Vögeler, “Delay attacks—Implication on NTP and PTP Time Synchronization,” IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, 2009, pp. 1-6.

[11] Q. Yang, D. An and W. Yu, “On Time Desynchronization Attack Against IEEE 1588 Protocol in Power Grid Systems,” *2013 IEEE Energytech*, pp. 1-5, 2013.

[12] E. Lisova, G. Marina, S. Wilfried, U. Elisabeth, Å. Johan, D. Radu and B. Mats, “Protecting Clock Synchronization: Adversary Detection Through Network Monitoring,” *Journal of Electrical and Computer Engineering, Hindawi*, vol. 2016, 2016.

[13] A. Treytl and B. Hirschler, “Securing IEEE 1588 by IPsec Tunnels-an Analysis,” IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, 2010, pp. 83-90.

[14] N. Moreira, J. Lázaro, J. Jimenez, M. Idirin and A. Astarloa, “Security Mechanisms to Protect IEEE 1588 Synchronization: State of the Art and Trends,” IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), 2015, pp. 115-120.

[15] R. Annessi, J. Fabini, F. Iglesias and T. Zseby, “Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization,” arXiv preprint, arXiv:1811.08569, 2018.

[16] T. Mizrahi, “A Game Theoretic Analysis of Delay Attacks Against Time Synchronization Protocols,” IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication Proceedings, 2012, pp. 1-6.

[17] T. Mizrahi, “Security Requirements of Time Protocols in Packet Switched Networks,” (No. rfc7384) 2014.