IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 27, 2021, accepted August 11, 2021, date of publication August 24, 2021, date of current version August 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3107203

A Security Management Architecture for Time Synchronization Towards High Precision Networks

HONGXING LI, DENGKUI LI, XIAODONG ZHANG, GUOCHU SHOU, YIHONG HU, AND YAQIONG LIU, (Member, IEEE)

Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing 100876, China
School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Guochu Shou (gcshou@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 92067102, and in part by the Project of Beijing Laboratory of Advanced Information Networks.

ABSTRACT Time synchronization is quickly becoming a fundamental prerequisite for a smart society. With the development of the fifth-generation (5G) network, time-sensitive networking (TSN), and the rise of high-precision networks, its accurate and reliable features have attracted an increasing amount of attention. As the most promising protocol with sub-microsecond accuracy, precision time protocol (PTP) has been widely used for network synchronization, and its proper operation and security are critical to the industries that build the infrastructure for a smart society. In order to provide synchronization security as a service, this paper presents a scheme based on software-defined networking (SDN) and network functions virtualization (NFV) principles for synchronization security. Security management is built as a virtual network function (VNF), and a mitigation mechanism is proposed to detect delay attacks and generate countermeasures. Finally, we investigate the impact of random delay attacks, constant delay attacks, and linear delay attacks and verify the performance of the proposed mitigation mechanism through experiments. The results show that the scheme is capable of detecting PTP delay attacks and mitigating their impact on time synchronization.

INDEX TERMS Time synchronization, delay attacks, precision time protocol, software-defined networking, network functions virtualization, time-sensitive networking, high precision communication.

I. INTRODUCTION

Time synchronization is the smart and digital society's foundation, as the infrastructure of the future society is full of distributed systems and connected by ubiquitous networks. Especially with the evolution of networks (e.g., the fifth-generation network, high precision networks) [1], [2], the need for the distribution of accurate and reliable synchronization is rapidly increasing. For example, Google's distributed database system requires data centers to maintain precise time synchronization for data consistency [3]. The European Securities and Markets Authority requires $100\mu s$ accuracy to achieve an undisputed 1ms transaction record for high-frequency financial trading [4]. New services like high-accuracy positioning and new technologies like intra-band contiguous carrier aggregation require ultra-high time accuracy in the fifth-generation (5G) network [5]. As the number of base stations is anticipated to be ten times the current scale, the transport networks are considered an effective solution to guarantee the accuracy goal of 130ns specified by the 3rd Generation Partnership Project (3GPP) [6].

Furthermore, the networks are expected to be more deterministic, providing not only bandwidth and reliability assurance but also time assurance (e.g., latency, jitter, timestamps for recording events, etc.). There will be an increasing number of time-engineered communications services, and networks must support time precision in data delivery as a fundamental communication service. TSN (time-sensitive networking), which prioritizes time synchronization as a key feature, is quickly becoming the standard for deterministic networking in local area networks [7]. The International Telecommunication Union (ITU) has proposed categorizing basic time-engineered services into in-time, on-time, and coordinated services, as well as externalizing time as a central property for future networks [8]. As a result, robust time synchronization for safety-critical applications with precise

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

timing requirements is becoming more prominent, as malicious attacks on time synchronization services can jeopardize the reliability of devices and services [9]. When the smart grid time synchronization service is attacked, for example, the distribution line may be physically damaged, or a large-scale power outage may occur [10].

We consider IEEE 1588 (which has now been updated to IEEE 1588-2019) [11], [12], also known as precision time protocol (PTP), in our investigation of the possibilities and implications of time synchronization breaching. It is widely used in distributed systems and is expected to be the most promising time synchronization protocol with ultra-high accuracy for future high-precision networks. Regarding the security of PTP, several types of threats are discussed in [13], including manipulating the content of the synchronization message, masquerading as the master clock, replaying of old messages, denial of service, and delay of synchronization messages. An optional secure extension of IEEE 1588 called Annex K [11] and some literature [13]–[15] have proposed approaches to counter the first four threats using cryptographic protocols, group source authentication, message integrity check, and message counter. When considering the artificially imposed asymmetric delay of synchronization messages, most synchronization algorithms are vulnerable, as they rely on measuring delays without taking adversary attacks into account [16]. As a result, the delay attack is a chronic misery to time synchronization and one of the most challenging threats to PTP. Attackers can impose delay attacks through man-in-the-middle attacks via Address Resolution Protocol (ARP) poisoning [17] or damaged switching devices [18]. Common network elements (NEs), such as switches, routers, etc., are likely to be targeted by attackers [19], who can force the attacked switches to tamper with the victim's traffic or redirect it in a way that compromises its confidentiality. Especially with the popularity of NFV, the software-defined NEs (e.g., Open vSwitch, vRouters, etc.) running on commercial servers make it more critical to prevent delay attacks for the synchronization security of high precision networks.

Software-defined networking (SDN) and network functions virtualization (NFV) technologies [20], [21] have developed rapidly in recent years, and future network infrastructures, including telecommunication networks, data centers, and operation networks, are evolving towards software-defined virtualized systems, using SDN/NFV technologies. In a traditional network, NEs use the routing protocol to determine how to forward data packets, whereas in an SDN/NFV enabled network, the routing decision is stripped from the NEs, and they are only responsible for collecting and reporting network status and processing packets according to the imposed forwarding rules from the control unit. As a result, the network becomes programmable and flexible, and the SDN controller can schedule traffic across the network at a global level. Simultaneously, SDN-based time synchronization and software-defined time synchronization networks are proposed and discussed [22]–[24]. The solution towards the synchronization security issues has opened up new possibilities. On the one hand, the software-defined time synchronization architecture simplifies the programming and collection of the synchronization status, including synchronization mode, synchronization state machine, and synchronization parameters, which can be used to monitor attacks. On the other hand, NFV enables it to build synchronization security as a smart and fast service based on virtual network functions.

Motivated by the issues and needs identified previously, we propose a security management scheme for attacks on time synchronization over packet networks. The main contributions are threefold as follows:

1) Towards the increasingly widespread and diverse security requirements of time synchronization, an SDN/NFV enhanced architecture is proposed to provide synchronization security as a service by building synchronization securities as virtual network functions (VNFs).

2) A mitigation mechanism is designed based on the proposed architecture to handle PTP delay attacks by coordinating the synchronization controller, network controller, and VNFs. The mitigation mechanism detects the attacks by jointing the network information and time synchronization status and mitigates their impacts by pre-planning new disjoint synchronization paths based on historical path data.

3) To investigate the influence of different PTP delay attacks and the performance of our proposed scheme, we analyze the characteristics of offset and path delay changes under random delay attacks, constant delay attacks, and linear delay attacks through experiments. Experimental results demonstrate the effectiveness of the proposed mitigation strategy.

The rest of this paper is organized as follows: Section II introduces the related work on synchronization attacks. Section III describes the SDN/NFV enhanced synchronization security management scheme, with the delay attack mitigation mechanism as an example. The experiments and results are presented in Section IV. Finally, Section V concludes this paper.

II. RELATED WORK

The security of time synchronization has caught the interest of researchers, and several efforts have been undertaken to improve it. Common PTP security aspects are investigated and analyzed in [10], [27]–[33]. Moussa *et al.* presented a review of the literature where security assessment of time synchronization mechanisms is targeted, addressed the security of time synchronization mechanisms in a smart grid environment, identified their gaps, and proposed mitigation strategies in [10]. Malicious attacks, including denial of service, Byzantine master, interruption of the control loop, removal of packets from the control loop, packet manipulation, packet insertion, and selective packet delay, are classified, and the corresponding security mechanisms are summarized in [27]. Treytl *et al.* described the security extension of IEEE 1588 and provided an analysis of the applicable

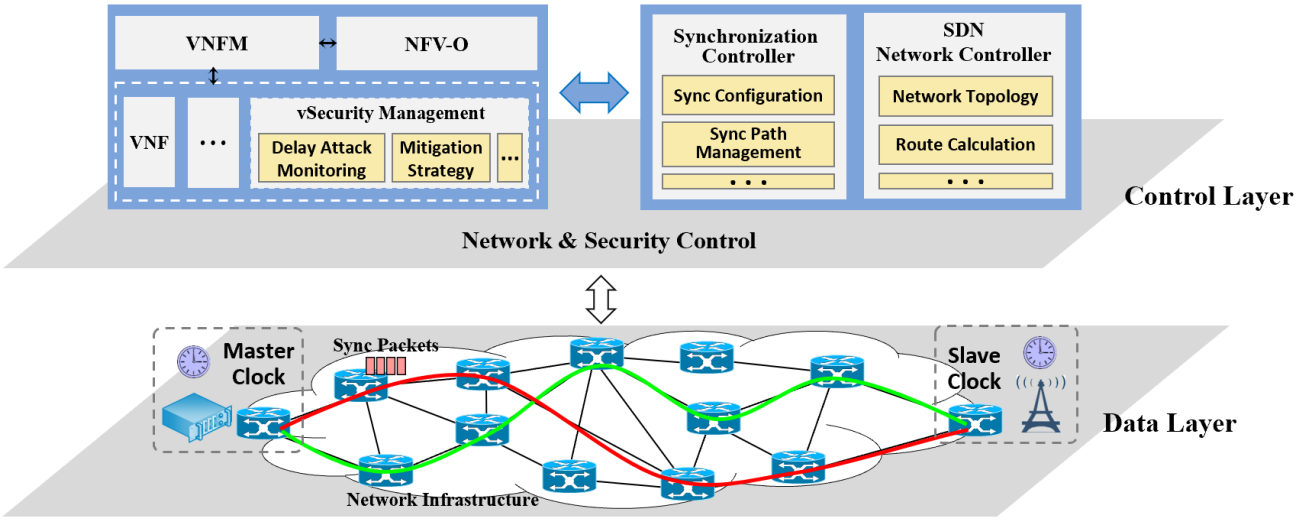

FIGURE 1. SDN/NFV enhanced network architecture against time synchronization attack.

threads as well as an attack targeting the master election algorithm in [28]. C. Onal *et al.* evaluated the sufficient and necessary aspects of IEEE 1588 Annex K's security mechanism and recommended modifications for authentication algorithms [29]. In [30], the authors summarized the various threats targeting time synchronization and divided them into two categories, either the slaves are aware of their desynchronized clocks after the attack or they are unaware. In [31], the authors analyzed security vulnerabilities of PTP caused by a flaw in the integrity check value calculation and elaborated countermeasures that protect protocol addresses and prevent the misuse of security associations to replay synchronization messages. Mizrahi *et al.* investigated PTP security solutions using two such protocols: Internet Protocol Security (IPsec) and Media Access Control Security (MACsec), characterized the typical deployment scenarios and performed a threat analysis of these scenarios in [32]. While Treytl and Hirschler [33] presented the impact of the use of IPsec tunnels and the precision of clock synchronization in PTP.

Regarding delay attacks, Ullmann M. *et al.* analyzed the impact of delay attacks on PTP that attackers disrupt time synchronization by changing the symmetry of link delay [34]. According to the characteristics of the delays added by attackers, Tal Mizrahi *et al.* classified PTP delay attacks into three types: constant delay attack, linear delay attack, and random delay attack. They introduced a method for analyzing delayed attacks using game theory models, analyzed possible strategies of attackers, and introduced a way to mitigate delay attacks using multiple paths between the master clock and the slave clock, which applies to specific systems with constant network delays [35]. Lisova E. *et al.* theoretically analyzed the indicator values affected by the attack and proposed a monitoring method based on the dynamic definition of decision thresholds [36], [37]. Through setting a dynamic sliding window, Lisova E. *et al.* determined whether the system is under delay attack based on the average value of the offset, the standard deviation, and other dynamic update thresholds in the sliding window. Robert Annessi *et al.* used the drift model to set an upper limit on the clock drift between subsequent synchronization signals, setting the threshold by the maximum clock drift rate and the timing signal's maximum and minimum propagation delay [38]. Lakshay Narula *et al.* focused on time synchronization in wireless networks, measuring the Round-Trip Time (RTT) of the link in advance and the RTT comparison obtained during synchronization to determine whether there is a delay attack [39]. Moussa B. *et al.* studied the delay attack on the power grid and proposed an algorithm to compensate for the impact of the delay attack or use the previously stored average instead of the attacked data to synchronize until the threat is released [40]. In [41], the authors proposed monitoring the path delay of the time synchronization and discarding the abnormal data packets when the path delay is higher than a certain threshold. Most current studies determine whether a delay attack is imposed by comparing the indicators obtained directly from the local clock with the set threshold, including the fixed and dynamic thresholds.

Unlike previous research, we focus on how to satisfy the increasingly widespread and diverse security requirements of time synchronization by providing synchronization security as a service. This paper presents how synchronization security could benefit from building the detection and mitigation of synchronization attacks as virtual network functions and jointing the network state information and the time synchronization data for delay attack mitigation.

III. SDN/NFV-BASED SECURITY MANAGEMENT FOR TIME SYNCHRONIZATION

A. NETWORK ARCHITECTURE

Fig.1 depicts the hierarchical architecture of the SDN/NFV enhanced network architecture against the security threats for

time synchronization. In general, the hierarchical architecture is designed to enable the security management of PTP attacks as a network function and implement the logically centralized control of the PTP attack mitigation strategy by jointing the network information with the synchronization information.

The detailed description of the architecture is presented as follows. The data layer of the architecture is made up of network devices and time synchronization devices like the master clock, network elements, and slave clock. Apart from the devices directly involved in the synchronization process, we also consider computing devices with computation and storage capacities that make up the cloud environment. Networking, computing, storage, and timing hardware make up the infrastructure of the architecture. The control & orchestration layer then abstracts the hardware resources of the data layer, allowing for logically centralized control, management, and orchestration of the virtual resources. The implementation of these functions depends on the following components:

- ***Network Controller***: which is a central entity that is aware of the network topology. It collects the NEs' and links' information using the network's global view. It calculates the route and configures the functions of network elements (e.g., flow table).
- ***Synchronization Controller:*** which is responsible for managing and configuring the synchronization functions. There are several functional modules: 1) The synchronization configuration module manages and configures the master clocks, such as the Grand Master (GM), Primary Reference Time Clocks (PRTC), and enhanced Primary Reference Time Clocks (ePRTC), etc. in the network. In addition, it generates a virtual synchronization network configuration strategy for synchronization path planning as well. 2) The synchronization path management module determines the synchronization path between the master and slave clocks, pre-configures alternate synchronization paths, and switches the synchronization path. Furthermore, the synchronization controller can support additional time synchronization functions, such as generating precision compensation strategies based on whether or not the network is PTP-aware [24].
- ***vSecurity Management:*** which is defined as a kind of virtual network function (VNF). It runs on virtual machines and consists of independent virtual security management functions (VSMFs), such as delay attack monitoring and mitigation strategy generation. Additionally, VSMFs are managed by the Virtualized Networks Function Management (VNFM), which is responsible for managing the VSMFs' life-cycle management. (e.g., initialization, suspension, and termination of the virtual machine).
- ***NFV Orchestrator (NFV-O):*** which is responsible for network controller, synchronization controller, and VSMFs coordination, as well as orchestrating the network time synchronization function, including synchronization path configuration, attack monitoring and mitigation, etc.

In the proposed architecture, security management and attack mitigation are built as virtual network functions, making it possible to provide security as a service for time synchronization with the assistance of the network controller and synchronization controller, as well as provide new countermeasures against synchronization attacks. For example, the centralized management of the synchronization controller is helpful in providing protection against modification and masquerading attacks by controlling which network devices in the domain can send synchronization messages (SYNC, DELAY_REQUEST, and DELAY_RESPONSE, etc.); the SDN-based synchronization path configuration and monitoring make it difficult to spoof or inject a malicious message into the domain. The following sections focus on the detailed security management scheme to monitor delay attacks and generate mitigation strategies to counter them.

B. PROBLEM DESCRIPTION

As the most promising way for precision time synchronization, PTP allows systems with clocks of various resolutions, precision, and stability to synchronize to a single time reference with sub-microsecond accuracy [25]. In a PTP system, the clocks are organized into a master-slave synchronization hierarchy. The synchronization is achieved by exchanging PTP timing messages with the slaves using the timing information to adjust their clocks to the time of their master in the hierarchy [11].

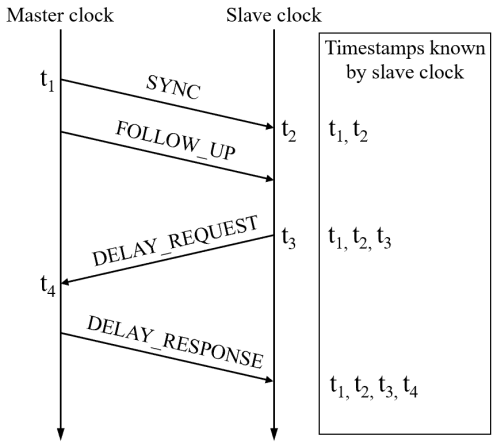

FIGURE 2. PTP synchronization mechanism.

The procedures of PTP, depicted in Fig. 2, are as follows:

1) The master clock periodically sends SYNC messages, including the egress timestamp t_1 to the slave clock. Optionally, it can also send a FOLLOW_UP message containing t_1.
2) When receiving the SYNC message, the slave clock records the timestamp as t_2.
3) The slave clock sends the DELAY_REQUEST message to the master clock, which includes the egress timestamp t_3.

4) The master clock records the ingress timestamp t_4 of the DELAY_REQUEST message and sends it back to the slave clock through a DELAY_RESPONSE message. Finally, the slave clock gets all four timestamps and computes the time offset from the master clock.

The offset and path delay between the given master and slave clocks are approximated by:

$$delay = \frac{(t_2 - t_3) - (t_1 - t_4)}{2}, \tag{1}$$

$$offset = \frac{(t_2 - t_1) - (t_4 - t_3)}{2}. \tag{2}$$

The slave clock adjusts its local clock based on the offset value calculated. When considering the implementation, some optimization mechanisms are commonly used to mitigate the impact of network jitter on time synchronization when updating the local time. For example, the clock rate ratio C_r is used to make errors resulting from differences in the rates of the master and slave clocks negligible for the accuracy required by the application. The moving median filter with a certain length is used to filter the raw data after the raw value of the path delay is calculated using Eq. (2), and the slave clock chooses the median value of a group of continuous raw values as the path delay $Delay_{filtered}$ to compute the final offset value. As a result, Eq (1) and (2) can be rewritten as:

$$delay = \frac{(t_2 - t_3) \times C_r - (t_1 - t_4)}{2}, \tag{3}$$

$$offset = t_2 - t_1 - delay_{filtered}. \tag{4}$$

The premise of Eq. (1), (2), (3), (4) is to assume that the delays of the forward and reverse directions are equal. As a result, it is vulnerable to packet delay attacks. PTP delay attacks can be classified into three types based on the characteristics of the delays added by the attackers: constant delay attacks, linear delay attacks, and random delay attacks [37].

- ***Constant delay attack:*** The extra delay imposed artificially by the attacker is the same all the time. Under the constant delay attack, the offset value between the master and slave clocks will mutate when the attack occurs and gradually tends to be stable. The path delay will suddenly increase and then remain at a high level. In mathematical terms, for an extra delay ed_i of a given synchronization cycle SC_i, a constant extra delay σ must be held:

$$\forall SC_i : ed_i = \sigma. \tag{5}$$

- ***Linear delay attack:*** The imposed artificial delay is increasing linearly and reaching a certain high-level σ for a long time. It is more difficult to detect because the attacker may impose a slight delay each time, accumulating a significant amount in the end. Following a linear delay attack, the offset value between the master and slave clocks will gradually change, and the path delay will grow linearly until it reaches a high-level state. Mathematically, for an increment step Δd and an extra delay ed_i of a given synchronization cycle SC_i, the following must hold for a linear delay attack:

$$\forall SC_i : ed_i = \begin{cases} i \cdot \Delta d & 0 \le t < T, \\ \sigma & t \ge T. \end{cases} \tag{6}$$

- ***Random delay attack:*** The attacker artificially imposes random delay values to bring a targeted salve clock into an unsynchronized state, disguised as the normal jitter of the network. After a random delay attack occurs, the offset value between the master and slave clocks will fluctuate significantly, and the path delay will change randomly. In mathematical terms, for a minimum extra delay, a maximum extra delay, and an extra delay ed_i of a given synchronization cycle SC_i, the following must hold for a random delay attack:

$$\forall SC_i : ed_{min} \le ed_i \le ed_{max}. \tag{7}$$

Note that a delay attack should be persistent since the PTP mechanism will calculate a correct offset, and the slave clocks will become synchronized again once a delay is not imposed anymore.

C. SECURITY MANAGEMENT SCHEME

1) GENERAL DESCRIPTION

Fig. 3 illustrates the operation of the proposed security management scheme. Communications among the network controller, synchronization controller, slave clock, and VSMF are shown in the illustration (see also Algorithm 1 below). To better understand the essential operations during a synchronization cycle, the proposed security management and control signal exchange process are detailed in the following:

- When a slave clock needs the security service for the exchange of PTP messages, it can send a time synchronization (TS) message to the synchronization controller, which is used to exchange the security service request and response between the slave clock and the synchronization controller, along with its identifiers *salve_clock_id*. Then, the synchronization controller extracts the *salve_clock_id* and schedules a time synchronization task $TASK_j$. After that, it sends a synchronization path query message (SP-Query) to the network controller, which is used to exchange network topology information to obtain the synchronization path from the network controller. The synchronization controller then orchestrates $TASK_j$ to the VSMF, which is responsible for delay attack monitoring and mitigation.
- After the $TASK_j$ is scheduled, the synchronization controller begins the disjoint synchronization path pre-planning operation. The network controller sends the network topology (*vnet*) to the VSMF when it receives the *sync path pre-planning signal*, and the VSMF calculates the alternate synchronization path P_{alt} using the historical synchronization path data.

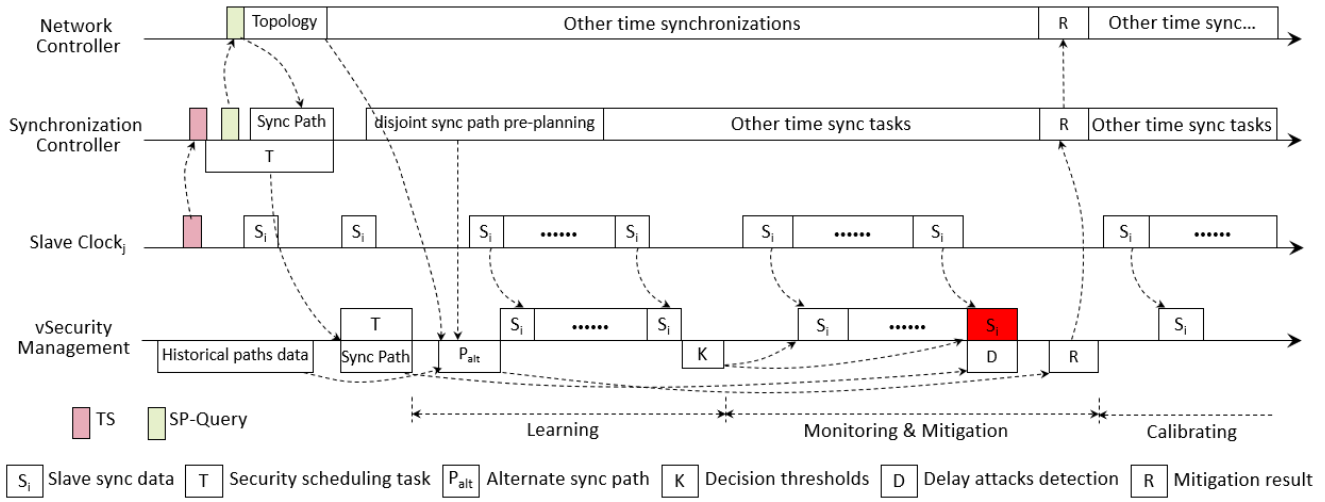

FIGURE 3. Illustration of the proposed security management scheme against PTP delay attacks.

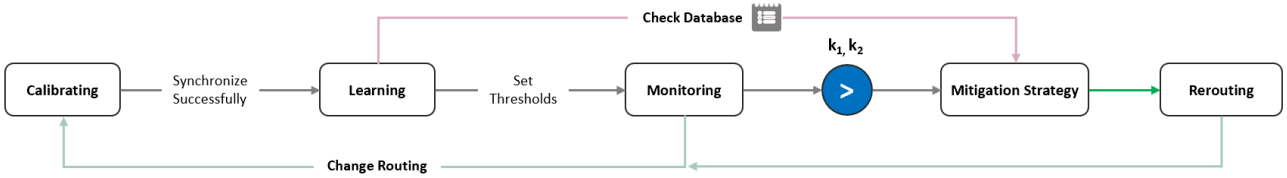

FIGURE 4. State transition of the proposed security management scheme.

2) ATTACKS DETECTION & MITIGATION DESCRIPTION

The slave clock can request the synchronization security service when it needs to be protected. Before the attack detection and mitigation take effect, the network state information (network topology, synchronization path, etc.) and synchronization information (slave clock id, synchronization state, etc.) should be gathered to the VSMF with the assistance of the synchronization controller and network controller. For PTP delay attacks, the synchronization states are divided into three phases, the synchronous calibration stage, the learning stage, and the monitoring & mitigation stage. Fig. 4 shows the state transition of the security management scheme, and the detailed description is as follows:

- During the calibration stage, the offset values of the slave clock will be relatively large at the beginning, as there is a considerable time deviation between the master and the slave clocks before synchronization. The offset value will then gradually decrease, and the VSMF will enter the learning stage once it receives the $TASK_j$.
- During the learning stage, the VSMF calculates the alternate synchronization path P_{alt}, and determines the threshold set K based on the offset and path delay values reported by the slave clock. Once the set K is acquired, it will enter the monitoring & mitigation stage.
- During the monitoring & mitigation stage, once the network topology changes, the network controller will synchronize the network information with the synchronization controller and the VSMF. As a result, the synchronization system will re-enter the calibrating stage. Otherwise, the VSMF will monitor the offset and path delay values reported by the slave clock. The VSMF will determine that the delay attack has been detected and send P_{alt} to the synchronization controller as the mitigation strategy if the threshold set K is broken and the synchronization path is not changed. Finally, the network controller makes the new path take effect.

After being processed by the proposed solution, the imposed PTP delay attacks will be detected by the VSMF, and the path between the master and slave clocks will be changed to an alternate secure one. As a result, the impact of the attacks will be mitigated. The following subsection presents detailed information on the PTP delay attack mitigation mechanism.

D. PTP DELAY ATTACKS MITIGATION MECHANISM

1) SYSTEM MODEL

The premise of correcting the slave clock by PTP is that the delays of the forward and reverse direction of the synchronization path between the master and slave clocks are equal. The attacker can use this vulnerability to launch an attack by increasing the delay of the forward or reverse direction to make the two directions' delays asymmetric. As a result, the synchronization accuracy is affected. We assume that the

Algorithm 1 Security Management Algorithm

1: Process executed at the Synchronization Controller
2: **if** (*TS-REQ_RCV=TRUE*) **then**
3: Extract the necessary information of $slave\_clock_j$
4: $TASK_j \leftarrow$ schedule a security task
5: Generate an NP-Query message with *slave_clock_id* and send it to the network controller
6: **if**(*NP-RESP_RCV=TRUE*) **then**
7: Get the path between the *master_clock* and $slave\_clock_j$
8: Send $TASK_j$ together with the synchronization path to the VSMF
9: Start disjoint synchronization path pre-planning
10: **else if** (*mitigation result received*) **then**
11: Update the synchronization paths blacklist database and send the virtual link to the network controller
12: Process executed at the Network Controller
13: **if** (SP-*Query_RCV=TRUE* || *routing changed*) **then**
14: Query the synchronization path between the *master_clock* and $slave\_clock_j$
15: Send the sync path to the synchronization controller
16: **else if** (*sync path pre-planning signal received*) **then**
17: Send the network topology *vnet* to the VSMF
18: **else if** (*mitigation result received*) **then**
19: Generate flow tables according to the virtual link and send them to the network elements
20: Process executed at the VSMF
21: **if** (*TASK_RCV = TRUE*) **then**
22: Extract the sync path and enter the learning stage
23: Determine the threshold set K
24: **else if** (*sync path pre-planningsignalreceived*) **then**
25: Get *vnet* from the network controller
26: Calculate the alternate synchronization path P_{alt}
27: **else if** (K!=NULL&& P_{alt}!=NULL) **then**
28: Enter monitoring & mitigation stage
29: **if** (*delay attacks detected*) **then**
30: Send P_{alt} to the synchronization controller
31: **end**

attacker adds extra delay ed_i to the forward direction. The path delay and offset under delay attacks can be rewritten as follows:

$$\overline{delay} = \frac{(t_4 - t_1) - (t_3 - t_2 - ed_i)}{2} = delay + \frac{ed_i}{2}, \quad (8)$$

$$\overline{offset} = \frac{(t_2 + ed_i - t_1) + (t_3 - t_4)}{2} = offset - \frac{ed_i}{2}. \quad (9)$$

According to Eq. (8) and (9), attackers will need an extra delay of T to add in one direction of the synchronization paths if they want to make a T/2 deviation between the master and slave clocks. Mathematically, when a delay attack occurs, the threshold of the offset value k_1 is given by

$$k_1 = T_{ac} - max\,|offset|\,, \quad (10)$$

where T_{ac} and $max\,|offset|$ denote the accuracy requirement of the application and the maximum value of the absolute offset recorded during the learning period, respectively.

We use a sliding window to monitor the path delay to reduce the impact of network load changes and control the rate of false triggers by instantaneous synchronization data. Firstly, the average path delay is calculated using a window W:

$$delay_{avg}^{W} = \frac{1}{W} \times \sum_{i=n}^{n+W} delay_i. \quad (11)$$

Then, the threshold of the path delay value k_2 is given by:

$$k_2 = 2k_1 + \sigma, \quad (12)$$

where σ denotes a small constant, which can be used to change the sensitivity of the decisions.

Let $offset_n$ and $delay_n$ denote the offset and path delay values for the nth round of time synchronization, respectively. The trigger conditions of clock information for delay attack detection are:

$$\left|offset_n\right| > k_1, \quad (13)$$

$$\left|delay_n - delay_{avg}^{W}\right| < k_2, \quad (14)$$

Finally, the set of thresholds K is given by:

$$K = \{k_1, k_2\}. \quad (15)$$

2) ALGORITHM SOLUTION

The mitigation algorithm is influenced by three major factors: network conditions, attacker strategies, and application accuracy requirements. In the SDN/NFV enabled network, the upper-layer VSMF can sense network topology changes through the network controller. When the delay changes due to network failure or routing changes, the VSMF can get the network information and recalculate the thresholds to avoid misjudgment caused by the change of the network status. The method of simultaneous monitoring of the offset and path delay values is adopted in the PTP delay attack mitigation algorithm. The detailed information is shown in Algorithm 2.

When the synchronization path is changed (line 1), the maximum value of offset $offset_{max}$ and the average value of the path delay $delay_{avg}^{W}$ are initialized to 0 (line 2). Then the synchronization cycle re-enters the calibrating stage (line 3). During the learning stage (line 4), the VSMF collects the synchronization information $S_j^{(n)}$ from $slave\_clock_j$, and updates $offset_{max}$ and $delay_{avg}^{W}$ (lines 5 and 6). At the end of the learning stage, the nodes on the current path are checked against the malicious path blacklist (lines 7). If none of the nodes is on the attacked synchronization paths marked in the malicious path blacklist database *PathAttackedData*, the set of the thresholds K is determined by Eq. (10) and (12) (lines 8-10). Otherwise, it means the attacker has already imposed attacks during the calibrating stage or learning stage. The VSMF will launch the mitigation strategy directly and

Algorithm 2 PTP Delay Attack Mitigation Algorithm

Consideration: time accuracy T_{ac}, synchronization information of slave clock $offset_n$, $delay_n$, current synchronization path $P_{current}$ and alternate path P_{alt}, malicious path blacklist database *PathAttackedData*.

1: **if** *routing changed* **then**
2: Initialization: $offset_{max} = 0,\ delay^{W}_{avg} = 0$
3: Enter calibrating stage
4: **else if** t **in** *the learning stage* **then**
5: Receive synchronization information from $slave\_clock_j$: $S_j^{(n)} = \{offset_n, delay_n\}$
6: Update:
$offset_{max} = max(|offset_{max}|, |offset_n|)$
$delay^{W}_{avg} = \frac{1}{W} \times \sum_{i=n}^{n+W} delay_i$
7: Get the malicious paths from the path blacklist database: pam = get(*PathAttackedData*)
8: **if** $P_{current} \cap pam = \emptyset$ **then**
9: $k_1 \leftarrow$ Use (10) to estimate the offset threshold
10: $k_2 \leftarrow$ Use (12) to estimate the path delay threshold
11: **else**
12: Send P_{alt} to the synchronization controller.
13: **else if** t **in** the *monitoring & mitigation stage* **then**
14: Monitoring the synchronization information of $slave\_clock_j$: $S_j^{(n)} = \{offset_n, delay_n\}$
15: **if** $|offset_n| > k_1$ & $|delay_n - delay^{W}_{avg}| < k_2$ **then**
16: **if** *routing unchanged* **then**
17: Send P_{alt} to the synchronization controller.
18: Enter calibration stage.
19: **end**

Algorithm 3 K-Shortest Alternate Path Algorithm

Input: *vnet*, *PathData*, $P_{current}$ and *PathAttackedData*
Output: P_{alt}

1: Get the malicious paths from the path blacklist database: pam = get(*PathAttackedData*)
2: **if** ($pam \cap$ find(*PathData*) = $\emptyset$& $P_{current} \cap$find(*PathData*) = [$master\_clock$, $slave\_clock_j$]) **then**
3: P_{alt} = find(*PathData*)
4: **else**
5: **for** *1:k* **do**
6: G = G – pam – $P_{current}$ – P_{alt}
7: $path$=**find-shortest-path**($master\_clock$, $slave\_clock_j$)
8: **if**(path != NULL) **then**
9: *PathData*=*PathData*+ [$master\_clock$, $slave\_clock_j$, $path$]
10: **if**($P_{current} \cap path$ = [$master\_clock$, $slave\_clock_j$])**then**
11: $P_{alt} = P_{alt} + path$
12: **continue**
13: **else**
14: disjoint constraint is not meet
15: **else**
16: there is no alternate sync path in *vnet*
17: **end**

send the alternate synchronization path P_{alt} to the synchronization controller (line 12). Afterwards, the system enters the monitor & mitigation stage (line 13). When the $offset_n$ and $delay_n$ exceed the threshold set K, and there is no routing change during the stage (line 14), the delay attack is detected (line 15), the P_{alt} is fetched and sent to the control plane (line 17).

The alternate synchronization path P_{alt} used in algorithm 2 (lines 12 and 17 of algorithm 2) can be prepared in the disjoint synchronization path pre-planning phase (line 9 of algorithm 1) using algorithm 3, named K-Shortest Alternate Path Algorithm. The proposed algorithm is combined with the historical synchronization path data *PathData* and the synchronization path blacklist data *PathAttackedData* to find the k-shortest synchronization paths as an alternate choice for each pair of master and slave clocks. As shown in algorithm 3, when receiving the disjoint synchronization path pre-planning task, the VTSF obtains the malicious paths from the path blacklist database *PathAttackedData*(line 1). Then the VSMF queries the historical synchronization path database *PathData*. If historical paths contain the given master and slave clocks and do not intersect with the marked paths in the *PathAttackedData* (line 2), the alternate paths P_{alt} are obtained (line 3). Otherwise, the algorithm moves on to the calculation of the k-shortest paths (lines 5-16). Firstly, the paths marked as attacked, the current synchronization path, and the alternate paths are unloaded from the virtual network G (line 6). Secondly, the shortest path between the master and the slave clocks is searched using the Dijkstra algorithm (line 7). When a new path is discovered, and the disjoint constraint is satisfied (line 8), it is written into the historical synchronization path database along with the master and slave clock information (line 9). Repeat steps 6-16 until the query number reaches k.

IV. DEMONSTRATOR

To investigate the impact of different PTP delay attacks and the performance of the proposed security management scheme, we built a testbed out of computing devices, network devices, clock devices, etc. As shown in Fig.5, the packet network comprises several SDN switches and two disjoint synchronization paths between the master and slave clocks. The controller & orchestrator unit is made up of several x86 servers that form an SDN/NFV environment where the SDN controller, synchronization controller, and VSMF module run. The SDN controller is in charge of all SDN switches, and the clocks can communicate with the synchronization controller. As access nodes, switch A and switch D should be backed up in an actual deployment scenario. The master clock equipped with a rubidium oscillator is locked to the GNSS (Global Navigation Satellite System). The slave clock,

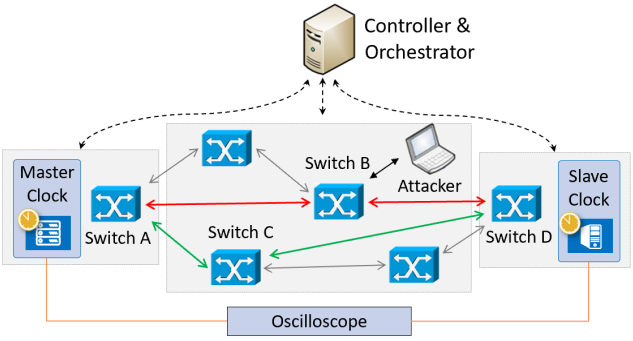

FIGURE 5. Experimental setup.

TABLE 1. Experimental configuration.

Equipment	Description
NFV Servers	Linux Host CPU: Intel(R) Xeon(R) CPU E3-1225 v3 @ 3.20GHz
SDN Controller	Linux Host Opendaylight-0.8.4
Master Clock	OSA 5421 Rubidium
Slave Clock	LinuxPTP CPU: Intel(R) Xeon(R) CPU E3-1225 v3 @ 3.20GHz
SDN Switch	CENTEC v350
Laptop	Lenovo Y410P

which runs the LinuxPTP application, communicates with the master clock via the built packet network under room temperature conditions that do not fluctuate drastically. The time synchronization cycle is set to 1 second. Both the master and slave clocks support hardware timestamp, whose resolution is 8ns under 125MHz hardware-driving frequency. In our experiment, the attacker imposes the delay attacks through SDN Switch B.

The detailed experimental setup is shown in Table 1. After the experiment begins, the synchronization path A-B-D is established under the control of the SDN controller, and the VSMF pre-configures A-C-D as the alternative synchronization path by interacting with the SDN controller and the synchronization controller. The offset and path delay values are uploaded from the slave clock to the VSMF module after the PTP message exchange begins.

A. GENERATION OF PTP DELAY ATTACKS

In the experiment, we assume that the simulated attacker gains control of SDN switch B via vulnerability attacks or brute force attacks and adds extra delay to the forward direction of the time synchronization path (primarily affecting the SYNC and DELAY_RESPONSE messages).

To generate the extra delay, we configure the flow table of SDN switch B to let the target messages go through a 20km optical fiber several times (indexed by the letter k) before forwarding to the next hop. The extra delay added by a single 20km optical fiber propagation is 100us with a system error of 2us. As a result, we can impose the three types of delay attacks according to Eq. (5), (6), and (7) by changing the value of k. The application requirement for time synchronization accuracy is set to 1.5us, and the attacker's target is to destroy the synchronization by adding 500us of extra delay in the forward direction. The detailed generation strategies are as follows:

- For the constant delay attack, we set $k = 5$, and the extra delay σ is 500us for each synchronization cycle after the attack imposed.
- For the linear delay attack, we set k increased from 1 to 5 linearly in 25 seconds and then kept it at 5 to generate an extra linear delay with an increment step size of 100us/5s after the attack imposed.
- For the random delay attack, we set k randomly belonging to [1,5] over time. As a result, the ed_{min} is 100us and the ed_{max} is 500us for each synchronization cycle after the attack imposed.

B. IMPACT OF DELAY ATTACKS ON PTP PROCESS

Fig.6 shows the offset values uploaded from the slave clock per second without attacks, which are kept within ±400ns. In other words, the offset baseline of the SDN switches used in this experimental environment is within ±400ns.

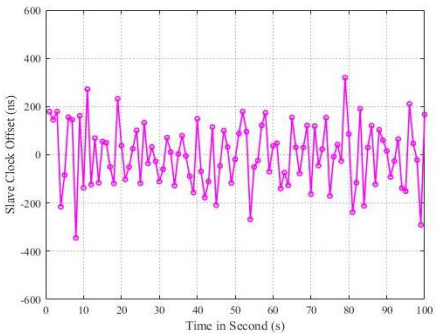

FIGURE 6. Offset value changes without attacks.

Then, we impose three different delay attacks on the time synchronization. In general, the measured changes of the slave clock offset value under various delay attacks are shown in Fig.7, and the path delay value calculated by the slave clock is shown in Fig.8. In the three delay attacks scenarios, the attacks are imposed at around the 23rd second. It can be seen in Fig.7 and Fig.8 that, after the attacks are imposed, the offset value will be abnormal in the next second, and the path delay value will change abnormally in about 5 seconds because of the moving median filter in LinuxPTP [42].

In terms of the offset value, the impacts of delay attacks are as follows: When the attacker adds a random extra delay to the forward direction of the synchronization path for continuous time, the offset values of the slave clock fluctuate continuously and randomly, and the variation value of the

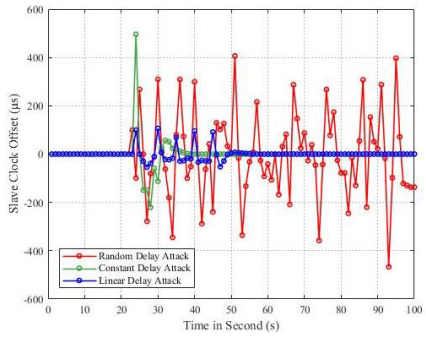

FIGURE 7. Offset value changes under delay attacks.

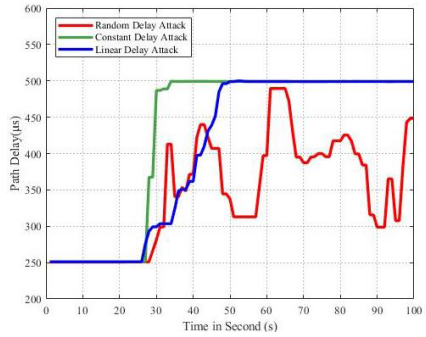

FIGURE 8. Path delay value changes under different delay attacks.

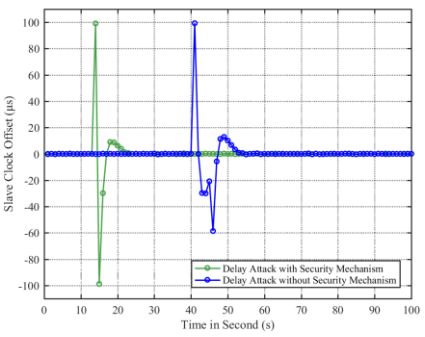

FIGURE 9. Offset value changes with delay attacks mitigation strategy or not.

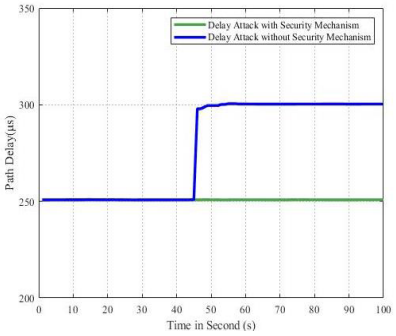

FIGURE 10. Path delay value changes with delay attacks mitigation strategy or not.

offset is related to the increase or decrease of the added delay, as shown by the red line in Fig.7. When a constant delay attack occurs, the offset value jumps to a high level at the point where the attack occurs and gradually decreases to a steady-state close to the situation where there is no attack, as shown by the green line in Fig.7. The linear delay attack causes multiple jumps in the offset value, with the offset value jump variation being smaller than the constant delay attack, as shown by the blue line in Fig.7. The offset value under delay attacks can be equal to the maximum of the imposed extra delay due to the use of filtered path delay when calculating the offset shown in Eq. (4). In terms of the impact of delay attacks on the path delay values, Fig.8 shows the changes in detail. The random delay attack causes the path delay to fluctuate over time, the constant delay attack causes it to jump from normal to high and then remain constant, and the linear delay attack gradually increases the path delay until it reaches a stable high level.

C. PERFORMANCE OF THE PROPOSED MITIGATION STRATEGY

Fig.9 and Fig.10 show the comparison of the offset value and delay path value with the proposed PTP delay attack mitigation strategy or not. In the comparative experiment, we impose a constant delay attack ($k = 1$) to the testbed and collect the offset and path delay values. The results are shown as the blue line in Fig.9 and Fig.10. Then, we enable the PTP delay attack mitigation strategy running in the controller & orchestrator unit and repeat the experiment. The results are shown as the green line in Fig.9 and Fig.10.

In both cases, the offset value produces an instantaneous transition at the point where the delay attack occurs. However, when the PTP delay attack mitigation strategy is enabled, the monitoring module monitors the time synchronization process with the assistant information (primarily synchronization path data) from the synchronization controller and SDN controller. Once a delay attack is detected, the mitigation module works with the synchronization controller and SDN controller to select a new synchronization path and switch the synchronization traffic to it. After the synchronization path is changed to A-C-D, the slave clock re-enters the synchronization calibration stage. During the process, the offset value curves are the same in both mitigated and unmitigated cases. However, the time error (phase difference between the master and slave clocks) curves are significantly different, just as shown in Fig.11. Without the security mechanism, the time error of the slave clock deteriorates to 50us, while with the security mechanism, it stays within ±200ns. The path delay cannot be monitored for changes throughout the process due to the moving median filter. As a result, the path delay value does not fluctuate dramatically, indicating that the attack has a minor impact on time synchronization under the security mechanism.

By this comparison, it can be seen that when there is no mitigation strategy in place, the attacker has successfully launched a delay attack, which increases the deviation

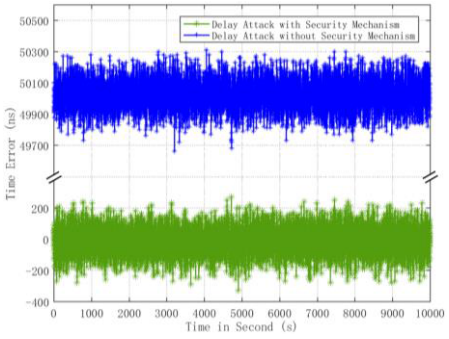

FIGURE 11. Time error changes with delay attacks mitigation strategy or not.

between the master clock and the slave clock by about 50us. After the PTP delay attack mitigation strategy works, the system successfully switches the synchronization traffic to the new security path to mitigate the attack's impact, which verifies the effectiveness of the proposed scheme.

V. CONCLUSION

The pace of digital and smart society is quickening, and time synchronization security will become increasingly important to the network. This paper investigates current research on synchronization attacks and corresponding protection schemes and proposes a software-defined security management scheme against PTP delay attacks based on SDN/NFV principles. The proposed scheme can generate an effective PTP delay attack mitigation strategy by joining network state information and time synchronization information. Experiment results verify that the scheme can effectively detect PTP delay attacks and mitigate their impact on time synchronization.

REFERENCES

[1] *Evolution Towards 5G-Advanced*, 3GPP, May 2021. [Online]. Available: https://www.3gpp.org/news-events/2194-ran_webinar_2021
[2] *Representative Use Cases and Key Network Requirements for Network 2030*, document ITU-T FG NET-2030, Jan. 2020.
[3] J. C. Corbett *et al.*, "Spanner: Google's globally distributed database," *ACM Trans. Comput. Syst.*, vol. 31, no. 3, Aug. 2013, Art. no. 8.
[4] *MiFID II-RTS 25: Clock Synchronization*, ESMA, Paris, France, 2017.
[5] H. Li, L. Han, R. Duan, and G. M. Garner, "Analysis of the synchronization requirements of 5G and corresponding solutions," *IEEE Commun. Standards Mag.*, vol. 1, no. 1, pp. 52–58, Mar. 2017.
[6] *Base Station (BS) Radio Transmission and Reception*, document TS 38.104, 3GPP, Sitges, Spain, 2020. [Online]. Available: https://www.3gpp.org/ftp//Specs/archive/38_series/38.104/38104-f80.zip
[7] IEEE 802.1 Working Group. (Oct. 2016). *Higher Layer LAN Protocols Working Group*. [Online]. Available: http://www.ieee802.org/1/
[8] *New Services and Capabilities for Network 2030-Description, Technical Gap and Performance Target Analysis*, document ITU-T FG NET-2030, Oct. 2019. [Online]. Available: https://www.itu.int/pub/T-FG-NET2030-2019-SUB.G2
[9] E. Lisova, M. Gutiérrez, W. Steiner, E. Uhlemann, J. Åkerberg, R. Dobrin, and M. Björkman, "Protecting clock synchronization: Adversary detection through network monitoring," *J. Electr. Comput. Eng.*, vol. 2016, pp. 1–13, Apr. 2016.
[10] B. Moussa, M. Debbabi, and C. Assi, "Security assessment of time synchronization mechanisms for the smart grid," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1952–1973, 3rd Quart., 2016.
[11] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Standard 1588-2008, 2008.
[12] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Standard 1588-2019, 2020.
[13] J. Tsang and K. Beznosov, "A security analysis of the precise time protocol (short paper)," in *Proc. ICICS*, Berlin, Germany, 2006, pp. 50–59.
[14] M. Bishop, "A security analysis of the NTP protocol version 2," in *Proc. ACSAC*, Dec. 1990, pp. 20–29.
[15] H. Krawczyk, M. Bellare, and R. Canetti, *HMAC: Keyed-Hashing for Message Authentication*, document RFC 2104, IETF, Feb. 1997.
[16] E. Lisova, E. Uhlemann, W. Steiner, J. Akerberg, and M. Bjorkman, "Game theory applied to secure clock synchronization with IEEE 1588," in *Proc. ISPCS*, Sep. 2016, pp. 1–6.
[17] 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*, 2016, pp. 872–878.
[18] M. Antikainen, T. Aura, and M. Särelä, "Spook in your network: Attacking an SDN with a compromised openflow switch," in *Proc. NordSec*, Cham, Switzerland, 2014, pp. 229–244.
[19] T. Sasaki, A. Perrig, and D. E. Asoni, "Control-plane isolation and recovery for a secure SDN architecture," in *Proc. NetSoft*, Jun. 2016, pp. 459–464.
[20] Open Networking Foundation. (2012). *Software-Defined Networking: The New Norm for Networks*. Accessed: Aug. 16, 2020. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf
[21] ETSI NFV ISG. (2012). *Network Functions Virtualization, White Paper*. Accessed: Aug. 16, 2020. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper.pdf
[22] S. Ruffini, P. Iovanna, M. Forsman, and T. Thyni, "A novel SDN-based architecture to provide synchronization as a service in 5G scenarios," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 210–216, Mar. 2017.
[23] L. Han, H. Li, L. Wang, and N. Hua, "A software-defined time synchronization solution in transport networks," in *Proc. OFC*, 2014, pp. 1–3.
[24] H. Li, G. Shou, Y. Hu, and Y. Liu, "SDN/NFV enhanced time synchronization in packet networks," *IEEE Syst. J.*, early access, Oct. 27, 2020, doi: 10.1109/JSYST.2020.3030664.
[25] *Time and Phase Synchronization Aspects in Packet Networks*, document ITU-T Rec. G.8271, 2012.
[26] M. Lévesque and D. Tipper, "Improving the PTP synchronization accuracy under asymmetric delay conditions," in *Proc. ISPCS*, Oct. 2015, pp. 88–93.
[27] G. Gaderer, A. Treytl, and T. Sauter, "Security aspects for IEEE 1588 based clock synchronization protocols," in *Proc. WFCS*, 2006, pp. 247–250.
[28] A. Treytl, G. Gaderer, B. Hirschler, and R. Cohen, "Traps and pitfalls in secure clock synchronization," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control Commun.*, Oct. 2007, pp. 18–24.
[29] C. Onal and H. Kirrmann, "Security improvements for IEEE 1588 annex K: Implementation and comparison of authentication codes," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control Commun.*, Sep. 2012, pp. 1–6.
[30] J.-C. Tournier and O. Goerlitz, "Strategies to secure the IEEE 1588 protocol in digital substation automation," in *Proc. CRIS*, Apr. 2009, pp. 1–8.
[31] A. Treytl and B. Hirschler, "Security flaws and workarounds for IEEE 1588 (transparent) clocks," in *Proc. Int. Symp. Precis. Clock Synchronization Meas., Control Commun.*, Oct. 2009, pp. 1–6.
[32] T. Mizrahi, "Time synchronization security using IPsec and MACsec," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control Commun. (ISPCS)*, Sep. 2011, pp. 38–43.
[33] A. Treytl and B. Hirschler, "Securing IEEE 1588 by IPsec tunnels—An analysis," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control Commun. (ISPCS)*, Sep. 2010, pp. 83–90.
[34] M. Ullmann and M. Vögeler, "Delay attacks-implication on NTP and PTP time synchronization," in *Proc. ISPCS*, Brescia, Italy, 2009, pp. 1–6.
[35] T. Mizrahi, "A game theoretic analysis of delay attacks against time synchronization protocols," in *Proc. ISPCS*, Sep. 2012, pp. 1–6.
[36] E. Lisova, E. Uhlemann, J. Åkerberg, and M. Björkman, "Delay attack versus clock synchronization—A time chase," in *Proc. ICIT*, 2017, pp. 1136–1141.
[37] E. Lisova, E. Uhlemann, J. Åkerberg, and M. Björkman, "Monitoring of clock synchronization in cyber-physical systems: A sensitivity analysis," in *Proc. IINTEC*, 2017, pp. 134–139.

[38] R. Annessi, J. Fabini, and T. Zseby, "SecureTime: Secure multicast time synchronization," 2017, *arXiv:1705.10669*. [Online]. Available: http://arxiv.org/abs/1705.10669
[39] L. Narula and T. E. Humphreys, "Requirements for secure clock synchronization," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 4, pp. 749–762, Aug. 2018.
[40] B. Moussa, M. Debbabi, and C. Assi, "A detection and mitigation model for PTP delay attack in an IEC 61850 substation," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 3954–3965, Sep. 2018.
[41] S. Ganeriwal, C. Pöpper, S. Čapkun, and M. B. Srivastava, "Secure time synchronization in sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 4, pp. 1–35, 2008.
[42] Network Time Foundation. (Aug. 12, 2008). *The Linuxptp Project Linuxptp V2.0*. [Online]. Available: http://linuxptp.sourceforge.net/

HONGXING LI received the bachelor's degree in communication engineering from Shandong University, China, in 2014. He is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, China. His research interests include time synchronization, time-sensitive networking, and edge computing.

DENGKUI LI received the B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, where he is currently pursuing the master's degree with the School of Information and Communication Engineering. His research interests include software-defined networks and time synchronization.

XIAODONG ZHANG received the bachelor's degree in electronic information science and technology from Inner Mongolia University, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, China. His research interests include time synchronization and time and frequency primary standard.

GUOCHU SHOU is currently a Professor with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His research interests include access network and edge computing, fiber and wireless network virtualization, network construction and routing, and mobile internet and applications.

YIHONG HU is currently an Associate Professor with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. Her research interests include fiber and wireless communications, network virtualization, and cloud computing.

YAQIONG LIU (Member, IEEE) received the double bachelor's degree in computer science and engineering and financial management from Tianjin University, China, in 2009, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2016. She is currently a Lecturer with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include data mining, networking and computing, spatial query processing, GIS, location-based services, and image animation.