applied sciences

MDPI

Article

Reference Broadcast-Based Secure Time Synchronization for Industrial Wireless Sensor Networks

Zhaowei Wang *, Dehua Sun and Chen Yu

School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China; 2212207066@stmail.ujs.edu.cn (D.S.); 2222007057@stmail.ujs.edu.cn (C.Y.)
* Correspondence: wangzhaowei@ujs.edu.cn

Abstract: Security is an important factor that cannot be neglected in the design of time synchronization algorithms since industrial wireless sensor networks are prone to attacks against physical nodes and communication links. The Sybil attack is an intelligent attack with a high destructive capacity in pretending multiple identities and broadcasting illegitimate messages to destroy the network operation. Existing secure time synchronization algorithms mostly focus on distributed protocols; however, they pay less attention to Sybil attacks and centralized network time synchronization. In this paper, we propose a novel reference broadcast-based secure time synchronization (RSTS) for industrial wireless sensor networks with a time source against Sybil attacks. Different from previous protocols, in converging the network structure and the clock status, RSTS employs a public neighbor forwarding mechanism based on reference broadcast to filter the illegal time information automatically. Instead of establishing a table with timestamps of packet transmission and receipt, the least square linear regression is utilized to estimate the compensation relative to the source node with the recorded time and calculated time difference in receiving packets. The simulation results demonstrate that RSTS is resilient to Sybil attacks as well as message manipulation attacks in comparison with existing algorithms.

Keywords: time synchronization; sybil attacks; security; industrial wireless sensor networks; message manipulation attacks

Citation: Wang, Z.; Sun, D.; Yu, C. Reference Broadcast-Based Secure Time Synchronization for Industrial Wireless Sensor Networks. *Appl. Sci.* **2023**, *13*, 9223. https://doi.org/10.3390/app13169223

Academic Editor: Guangquan Xu

Received: 15 July 2023
Revised: 10 August 2023
Accepted: 11 August 2023
Published: 14 August 2023

1. Introduction

Industrial wireless sensor networks (IWSNs) [1,2] connect sensors and actuators in a distributed manner, which is the important perception layer of the Industrial Internet of Things. Many fundamental applications in industrial networks, e.g., data collection [3], timely monitoring [4], time-division multiplexing communication [5], and other coordinated control operations [6], require the nodes in IWSNs to capture the production environment conditions and control the target device in a collaborative and synchronous manner, namely, time synchronization. Unfortunately, each sensor node records its time based on different crystal oscillators caused by manufacturing engineering and environmental influence [7,8], which results in asynchronous time. As time synchronization is crucial for IWSNs to run normally, many algorithms and protocols are proposed to improve the synchronization accuracy, enhance the robustness, and reduce the energy consumption in IWSNs under no attacks [3,4,7–11]. However, it ignores the fact that secure information acquisition is a key aspect in achieving various time synchronization behaviors.

The security problem is vital for the protocol design of IWSNs [12]. The openness of wireless communication and unattended deployment environment make IWSNs particularly vulnerable to intentional attacks [13,14], such as the physical attack against network devices and the delay attack against communication links [15,16]. In addition, time synchronization design involves two basic elements: timing and time information transmission, which happens to be closely related to the physical node attack and the delay attack on

IWSNs. Consequently, time synchronization is prone to attacks against node identity, communication links, and packets [14]. Complex security mechanisms, such as encryption and authentication, can effectively improve the algorithms' security [17,18] but are not suitable for IWSNs with severely limited resources including computing, communication, storage, and energy. Therefore, it is necessary to design a secure synchronization protocol in considering the characteristics of the IWSNs' time transfer system.

To meet the security requirements of time synchronization, many security mechanisms are proposed. Information encryption [18], authentication technology [17], and threshold detection [15,16,19,20] are the most commonly used defense methods. However, the research seldom considers intelligent attacks, such as the Sybil attacks [21,22], or the situation in that attackers collude with each other.

The Sybil attack is an intelligent attack mode, where the attackers can illegally pretend multiple identities [21,22]. In time synchronization, a Sybil attacker injects incorrect time information into the network with a disguised legal identity, which produces more serious disruption. It may mislead the valid nodes out of synchronization and be isolated. In addition, most structures of the perceptual control networks in industrial circumstances are centralized. The action of the lower node depends on the command from the master node, namely, the lower node needs to synchronize with the master node, which is termed a reference time source in the time synchronization system. In conclusion, centralized time synchronization is better suited for industrial applications. Unfortunately, the serious single point of failure problem is an unavoidable defect for centralized protocols, and the problem will be magnified under Sybil attacks. Information encryption and decryption can defend well against multiple types of attacks including the Sybil attack, but produce additional communication, computing, and storage costs, which are not desirable in resource-constrained IWSNs.

Consequently, we propose a reference broadcast-based secure time synchronization (RSTS) protocol to deal with intelligent attacks for centralized IWSNs. The core content of the RSTS protocol is to construct a novel time information transmission mechanism. RSTS aims to obtain a valid time difference in receiving packets from the same sender. Unlike the direct communication between master and slaver in existing protocols, the slaver in RSTS employs the public neighbor to acquire the time of the master indirectly. Subsequently, based on the recorded time and calculated time difference in receiving packets from the public neighbor, a least square linear regression is utilized to estimate the compensation relative to the master node. Hence, RSTS can filter the fake time information automatically. The main contributions of this study are summarized as follows:

(1) We propose a novel protocol RSTS aimed at improving the security of time synchronization in centralized IWSNs. A novel time information transmission mechanism is designed to obtain reliable time information with the verification of packet sequence numbers.
(2) RSTS employs the time difference when the master node and slave one receive packets from the public neighbor to block the interference of malicious messages from manipulators and masqueraders automatically instead of coping with large amounts of data to filter illegal messages or verify the node identity.
(3) We provide an effectiveness analysis and conduct simulations to verify the feasibility and efficiency of RSTS. The simulation results demonstrate that RSTS can defend against both Sybil attacks and message manipulation attacks.

The rest of the paper is organized as follows. Section 2 describes the related work of secure time synchronization protocols. In Section 3, we introduce the network model, clock model, attack model, and problem formulation. The RSTS protocol is detailed in Section 4, including protocol design, multi-hop network synchronization, security analysis, and communication energy cost. The simulations to assess the performance of the RSTS protocol are represented in Section 5. Finally, Section 6 concludes the article.

2. Related Work

Currently, the implementation of time synchronization mostly adopts centralized [23,24] or distributed manners [9,10]. In centralized ones, a reference node is selected and spreads its time to synchronize other network nodes rapidly. In contrast, distributed approaches do not require a specific reference node, and the local node updates its clock status robustly according to the neighbor's information. Furthermore, many time synchronization approaches have been studied in the literature for different requirements, e.g., precision [25,26], consumption [7,8,27], robustness [4,9,10], security [14], and so on. For any requirements, security is essential for sophisticated time information acquisition. Threshold detection [15,16,19,20], message filtering [21,22], and encryption [17,18] techniques are common defense measures in secure time synchronization. We present the review of the secure time synchronization methods as follows.

2.1. Threshold Detection

Normally, the link delay, offset, and relative clock skew between two nodes are stable and bounded in benign environments, and attackers usually broadcast random time information to disrupt the synchronization balance. Accordingly, an attacker can be distinguished by detecting the difference in packet transmission delays, clock offset, or relative clock skews exceeding the statistical threshold or not.

Qiu et al. [19] utilized the threshold detection technique to determine whether the received messages from the node's farther node are valid or not. Each node first calculates the clock offset between its father node and grandfather node. Based on the pretested threshold value of offset, each node determines whether the synchronization reference node is its father node or grandfather node. Furthermore, a spanning tree structure is constructed against fake timestamps for secure time synchronization. Jia et al. [20] divided the synchronization network into several clusters by the adaptive threshold-based K-means clustering algorithm. It should be noted that there is a chief cluster head (CCH) as the standard time source in the clustered network. The predefined threshold about the difference of varying rates of skew (VRS) between each node and the CCH is related to the clock quality. The clock quality varies between clusters, which leads to different synchronization frequencies between clusters, unlike simultaneous synchronization in most algorithms. Subsequently, a threshold-based two-tier fault detection algorithm is developed to overcome malicious attacks during the synchronization process.

Aimed at secure distributed time synchronization, He et al. [15] developed a secure average-consensus-based time synchronization protocol (SATS) against message manipulation attacks. SATS includes two checking processes for the hardware clock and the logical clock. Unlike isolating attack nodes in other safeguard mechanisms, SATS can flexibly utilize attack information to improve convergence speed in logical clock synchronization. Similarly, a secure maximum-consensus-based time synchronization protocol (SMTS) is further proposed in [16]. SMTS designs the logical clock-checking process based on the property that the local node can estimate its neighbor's correction parameters. The hardware clock-checking processes in SATS and SMTS determine whether a node is legal or not by comparing whether two consecutive estimated relative hardware skews are equal or not.

2.2. Message Filtering

In threshold detection-based secure synchronization methods, a suspicious node would be isolated once it is voted as an attacker. These methods are unfavorable for masquerade attacks. The message filtering technique sifts out the trusted messages to compute clock parameters rather than isolating the suspicious ones crudely. Linear clock characteristic promotes the design basis of message filtering based secure time synchronization.

Dong et al. [21] proposed RTSP to defend against the Sybil attack. As the time messages from the same node obey a linear relationship, namely, a conformance relationship, RTSP employs a graph theoretical approach to filter suspicious messages in the buffer

instead of identifying the node. Finally, the anomaly detection process is evolved into a dynamic programming approach to find the maximum clique, namely, valid time messages. To further reduce the security defense costs, Wang et al. [22] utilized the timestamp correlation among different nodes and the uniqueness of a node's clock skew to detect invalid information rather than isolating suspicious nodes. Although the two protocols [21,22] can defend against Sybil attacks, they are applied to fully distributed networks and are not suitable for centralized networks with a time source.

2.3. Encryption and Others

Encryption techniques can defend against various attacks principally in public key cryptography. However, secure key storage, encryption, and decryption operations may bring more storage, computing, and communication overhead. Du et al. [17] developed an authentication scheme to ensure secure time synchronization in heterogeneous wireless sensor networks. Rahman et al. [18] utilized pairing-based cryptography by computing the shared secret keys to secure the synchronization processes, with low communication and storage costs.

Due to the diversity of time synchronization methods, other security defense techniques have also been proposed. Moussa et al. [28] proposed an extension to the precision time protocol (PTP) to defend against attacks on the grand master clock, network, transparent clock, and slave clocks. The extension deploys a redundant clock as the network time reference to detect the synchronization of slave clocks. As a consequence, a feedback closed loop security-aware PTP is achieved. Sun et al. [29] provided redundant ways and multiple time sources for each node to synchronize with the time source. Each node computes the source clock offsets with all its neighbors and chooses the median as the source clock offset. The developed methods can be resilient to message missing and distortion caused by the malicious nodes, and each node with l normal neighbors can tolerate up to $l - 1$ malicious nodes.

In dynamic and hostile environments, node mobility and malicious nodes are unavoidable and have a serious effect on the averaged-based consensus time synchronization protocols. Phan et al. [30] proposed a neighbor-aware time synchronization protocol (NTSP) to overcome the limitations of gradient time synchronization protocol (GTSP). NTSP first adopts a marking technique to classify the neighbor's status for each node as synchronized, new, or unsynchronized. Secondly, depending on the number of synchronized neighbors, each node calculates the clock compensations including only synchronized neighbors or whole neighbors. The classification and calculating rules render NTSP more robust.

However, most countermeasures for various attacks seldom consider Sybil attacks. Additionally, due to the inherent vulnerability of topology, centralized time synchronization methods are more vulnerable to disruption under Sybil attacks. The child nodes take their parent nodes as their time sources, and then the branch of the parent nodes will be disabled when the parent nodes are attacked. Although encryption and authentication techniques are effective [31,32], additional computing and communication overhead cannot be popularized for the design of time synchronization protocol in resource-constrained IWSNs. Different from existing secure synchronization methods, our method automatically filters the adverse impact of attackers in utilizing the public neighbor broadcasting mechanism.

3. System Models and Problem Analysis

Maintaining the security of time synchronization in centralized IWSNs with low overhead is not a trivial task. In addition to various attack types, the unique network structure with multi-hop and time source has also increased the pressure of security defense on time synchronization. In this section, we introduce the network model, clock model, and attack model and further analyze the problem.

3.1. Network Model

We consider an IWSN with a reference node or root node. Each node has a unique identity. The ordinary node communicates with the reference node in a multi-hop manner, namely, the IWSN with the reference node is a layered network, as shown in Figure 1. Obviously, for an ordinary node, it has at least one parent node and one child node. However, the edge nodes only have parent nodes and the reference node only has child nodes. Although the layered network topology is simple and has a high transmission efficiency, it remains a practical problem that attacking the node may result in the failure of network branches starting from this node.

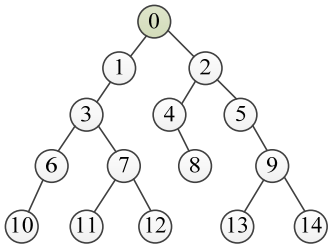

Figure 1. Schematic illustration of a layered IWSN. Node 0 is the reference node.

3.2. Clock Model

In general, the node's time is measured based on counting the output pulse of an equipped crystal oscillator. Furthermore, the clock reading of each node i is modeled as a first-order dynamic function with reference to the absolute time t, i.e.,

$$\tau_i(t) = \alpha_i t + \beta_i, \tag{1}$$

where α_i is the clock skew that determines the timing rate and β_i is the initial clock offset. Since the frequency of the oscillator determines α_i, each α_i is slightly different due to the imperfect oscillators, ambient temperature, battery voltage, and oscillator aging [15]. Hence, α_i satisfies $1 - \rho \leq \alpha_i \leq 1+\rho$, where ρ is the clock drift and typically in the range of $\left[10^{-5}, 10^{-4}\right]$ [9]. Even if β_i is equal at the starting moment, the time between nodes is still out of synchronization with the existence of α_i. In addition, the absolute t is unavailable, and it is impossible to compute α_i and β_i directly.

In order to maintain synchronization, it is crucial to find the relationship between the two nodes' clocks and the clock compensation parameters. Observing the clock readings of two nodes, it is indicated that

$$\tau_j(t) = \alpha_j\left(\frac{\tau_i(t) - \beta_i}{\alpha_i}\right) + \beta_j = \alpha_{ij}\tau_i(t) + \beta_{ij}, \tag{2}$$

where $\alpha_{ij} = \alpha_j/\alpha_i$ and $\beta_{ij} = \beta_j - \alpha_{ij}\beta_i$ denote the relative clock skew and offset, respectively. The goal of synchronizing with node j for node i is to find the compensation parameters α_{ij} and β_{ij}. Meanwhile, each node only synchronizes with its parent node. After multi-hop iteration, the local node achieves synchronization with the reference node.

3.3. Attack Model

Unattended operation and wireless communication make IWSNs vulnerable to various attacks, e.g., replay attacks, delay attacks, dos attacks, modifying and dropping timestamps, masquerade attacks, message manipulation attacks, etc. Referring to the definition in [15,16], the message manipulation attacker can pretend to be a safe node and broadcast unregulated incorrect time information, including repeatedly broadcasting the same timestamp, increasing transmission delay, and injecting error time. A masquerade attacker, also known as a Sybil attacker, illegitimately pretends to be another node and disrupts the time synchronization process. Hence, security attacks on time synchronization can be simply divided into message manipulation attacks and Sybil attacks.

Currently, there are various studies on the defense of message manipulation attacks. Although threshold detection is an effective security mechanism, it performs poorly under Sybil attacks. Accordingly, in this paper, we mainly focus on the Sybil attacks. For example, a Sybil attacker A can send out a disguised clock reading, i.e.,

$$\tau_A^i(t) = \alpha_A t + \beta_A + \omega_A(t), \tag{3}$$

where $\omega_A(t)$ denotes the attack power and i is the disguised node identity (or suspicious node).

It should be noted that IWSNs usually adopt medium access control or a digital signature to verify the message. Local nodes cannot arbitrarily modify the received message from other nodes [16]. This is an important assumption for our next design of the RSTS protocol.

3.4. Problem Analysis

With the example of a typical time synchronization algorithm, i.e., flooding time synchronization protocol (FTSP) [23], in this section, we perform a security analysis of the FTSP in layered IWSNs. From Equation (2), it can be observed that node i could estimate the compensation parameters related to its reference node j with the linear relationship among two clocks. In the FTSP, a linear regression table of sending timestamp $\tau_j(t)$ and receiving timestamp $\tau_i(t)$ is established; the estimation of α_{ij} and β_{ij} are as

$$\begin{cases} \hat{\alpha}_{ij} = \frac{\sum_{k=1}^{n}\left(\tau_j(k)-\overline{\tau_j}\right)\left(\tau_i(k)-\overline{\tau_i}\right)}{\sum_{k=1}^{n}\left(\tau_i(k)-\overline{\tau_i}\right)^2} \\ \hat{\beta}_{ij} = \overline{\tau_j} - \hat{\alpha}_{ij}\overline{\tau_i} \end{cases}, \tag{4}$$

where n is the size of the regression table and $\overline{\tau}$ is the mean. Normally, the reliability of message pairs $<\tau_i(k), \tau_j(k)>$, $k = 1, 2, \cdots, n$ is a prerequisite for accurately estimating α_{ij} and β_{ij}. If an attacker exists, the anomalous sending timestamp $\tau_A^j(t) = \tau_A(t) + \omega_A(t)$ hidden in the regression table would interfere with the estimation process. For message manipulation, the attacker may only affect the synchronization accuracy. Specifically, the Sybil attacker will mislead the node with more serious destructiveness.

To clearly describe the performance of FTSP under various attacks, we perform a simple simulation analysis on a chain IWSN with 5 nodes, and node 0 is the root. The hops between the normal nodes and the reference node are 1, 2, 3, and 4. The hop number and node identifier are the same. We assume that node 2 is a manipulation attacker or disguised by a Sybil attacker. Figure 2 shows the performance of safe nodes in FTSP under attacks. It can be clearly observed that the synchronization accuracy seriously decreases under manipulation attacks, and the clock offset of nodes behind the attackers would diverge under Sybil attacks.

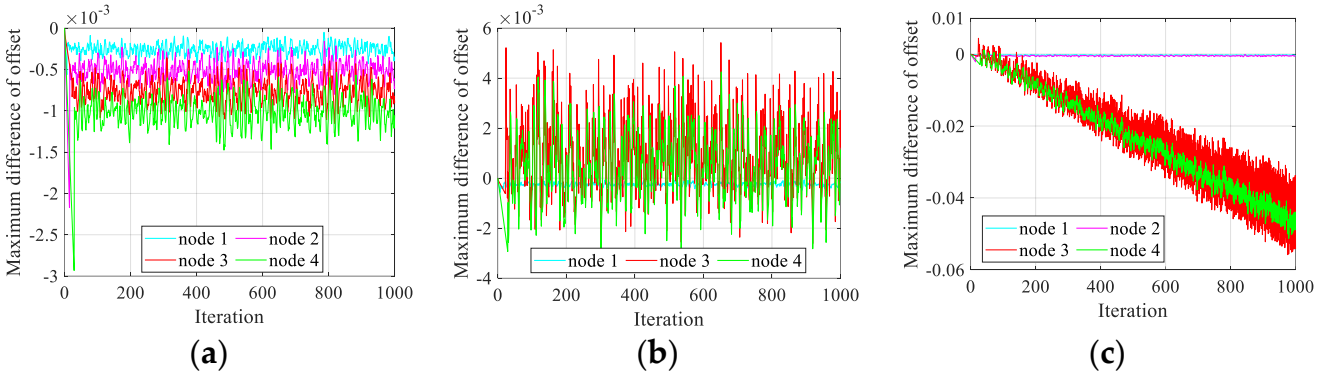

Figure 2. Performance of FTSP under various attacks. (**a**) No attacks; (**b**) manipulation attacks; and (**c**) Sybil attacks.

4. RSTS Protocol

In this section, we will describe the core idea of the RSTS protocol with a brief network model. Multi-hop network synchronization is then detailed, followed by performance analysis.

4.1. RSTS Synchronization

From Equations (2) and (4), under security attacks, the safe nodes use an invalid timestamp $\tau_A^j(t)$ for relative clock skew and offset updating, that is to say, the timestamps from attackers directly act on the estimation of compensation values. By analyzing Equation (1), the clock offset between two nodes i and j at time t is

$$\Delta\tau_{ij}(t) = \tau_j(t) - \tau_i(t) = (\alpha_j - \alpha_i)t + (\beta_j - \beta_i). \tag{5}$$

Note that $t = (\tau_i(t) - \beta_i)/\alpha_i$, which yields

$$\begin{aligned} \Delta\tau_{ij}(t) &= (\alpha_j - \alpha_i)\tfrac{\tau_i(t)-\beta_i}{\alpha_i} + (\beta_j - \beta_i) \\ &= (\alpha_{ij} - 1)\tau_i(t) + (\beta_j - \alpha_{ij}\beta_i) \\ &= (\alpha_{ij} - 1)\tau_i(t) + \beta_{ij} \end{aligned} . \tag{6}$$

Hence, based on messages $<\tau_i(k), \Delta\tau_{ij}(t)>$, $k = 1, 2, \cdots, n$, the estimation of α_{ij} and β_{ij} are rewritten as

$$\begin{cases} \hat{\alpha}_{ij} = 1 + \frac{\sum_{k=1}^{n}\left(\Delta\tau_{ij}(k) - \overline{\Delta\tau_{ij}}\right)\left(\tau_i(k) - \overline{\tau_i}\right)}{\sum_{k=1}^{n}\left(\tau_i(k) - \overline{\tau_i}\right)^2} \\ \hat{\beta}_{ij} = \overline{\Delta\tau_{ij}} - (\hat{\alpha}_{ij} - 1)\overline{\tau_i} \end{cases} . \tag{7}$$

Comparing Equations (4) and (7), presumably, we ask whether there are any reliable methods to obtain the valid $\Delta\tau_{ij}(t)$ under attacks. Classic reference broadcast synchronization (RBS) protocol [24,26] exchanges receiving timestamps of messages from a common reference node between two receivers. Each receiver computes its time offset to any other receiver as the average of the time offsets. Finally, synchronization is achieved among receivers rather than synchronizing with the reference node. Inspired by RBS, we design the RSTS protocol.

The key insight of RSTS is to exploit the public neighbor forwarding mechanism so that the illegal timestamps can be filtered. We summarize the RSTS process in Algorithm 1, where G denotes the topology graph of an IWSN, V is the set of nodes, e_{ij} indicates that node i and j can communicate, and N_i is the neighbor set of node i. In RSTS's synchronization process, the public reference neighbor j of two non-adjacent nodes l and i, i.e., the slave node l and the master node i, broadcasts its local time $\tau_j(t)$ marked by a preset sequence number $seq_j(t)$ periodically. The two nodes record the times based on their clocks once receiving the broadcast messages, i.e., $<\tau_i(t),\ seq_j(t)>$ and $<\tau_l(t),\ seq_j(t)>$. Subsequently, if the master node i is in the sync state, it immediately sends the timestamp $<\hat{\tau}_{i0}(t), seq_j(t)>$, where $\hat{\tau}_{i0}(t)$ is the estimated clock source time, to the slave node through public neighbor forwarding. Based on the n records $<\Delta\tau_{l0}(t), \tau_l(t)>$, where $\Delta\tau_{l0}(t)$ is the time difference of $\hat{\tau}_{i0}(t)$ and $\tau_l(t)>$, the slaver l estimates the clock compensation $<\hat{\alpha}_{l0}, \hat{\beta}_{l0}>$ in Equation (7). Meanwhile, node l achieves synchronization with the time source and denotes it as the sync state. From the above, detailed analysis of RSTS, it has a time complexity with $O(n)$. Since no node can manipulate the information received from its neighbors with information checking in medium access control protocol [16], the receiving timestamps from the master node can be safely forwarded through the public neighbor.

We use an example to illustrate how RSTS works. Consider a simple network scenario in Figure 3, where node m is the reference master node, node s is the slave node, node r is the shared neighbor node, and node A_i is the attacker. Under the benign environment, nodes m and s could receive messages from node r with valid sending timestamps. The relationship between timestamps is shown in Table 1.

Algorithm 1 RSTS protocol.

Input: $G = (V, E)$, $\hat{\alpha}_{i0}(0) = 1$, $\hat{\beta}_{i0}(0) = 0$, T, $\forall\, i, j \in V$, $e_{ij} \in E$, $j \in N_i$.
Output: $\hat{\alpha}_{i0}$, $\hat{\beta}_{i0}$, $\forall\, i \in V$.

1. $\forall\, j \in V$, if $\tau_j(t) = kT, k \in N^+$, node j broadcasts its time $\tau_j(t)$ with a random packet sequence $seq_j(t)$.
2. Upon receiving time information from node j, nodes i, l records $<\tau_i(t),\ seq_j(t)>$ and $<\tau_l(t),\ seq_j(t)>$, $i, l \in N_j$, respectively.
3. If node i is in the sync state, it broadcasts the sync time, where

$$\hat{\tau}_{i0}(t) \leftarrow \hat{\alpha}_{i0}(t)\tau_i(t) + \hat{\beta}_{i0}(t).$$

4. Node j forwards the sync packet received from node i to node l, $l \in N_j$, $l \notin N_i$.
5. If the $seq_j(t)$ matches, node l calculates $\Delta\tau_{l0}(t) = \hat{\tau}_{i0}(t) - \tau_l(t)$, and stores $<\Delta\tau_{l0}(t), \tau_l(t)>$.
6. Upon storing n records, node l estimates $< \hat{\alpha}_{l0}, \hat{\beta}_{l0}>$ as Equation (7) and then denotes it as the sync state.

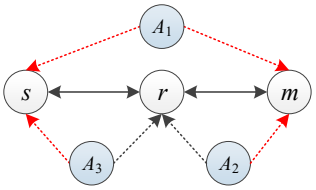

Figure 3. An illustrative example to show how RSTS works.

Table 1. An illustrative example to show how RSTS works.

Sending Timestamps in r	Attacker A_1	Attacker A_2	Attacker A_3	Receiving Timestamps in m	Receiving Timestamps in s	Matching
$\tau_r(1)$	—	—	—	$\tau_m^r(1)$	$\tau_s^r(1)$	Y
—	—	$\tau_{A_2}^r(1)$	—	$\tau_m^{A_2}(1)$	—	N
—	$\tau_{A_1}^r(1)$	—	—	$\tau_m^{A_1}(1)$	$\tau_s^{A_1}(1)$	Y
$\tau_r(2)$	—	—	—	$\tau_m^r(2)$	$\tau_s^r(2)$	Y
—	$\tau_{A_1}^r(2)$	—	—	$\tau_m^{A_1}(2)$	$\tau_s^{A_1}(2)$	Y
—	—	$\tau_{A_2}^r(2)$	—	$\tau_m^{A_2}(2)$	—	N
$\tau_r(3)$	—	—	—	$\tau_m^r(3)$	$\tau_s^r(3)$	Y
—	—	—	$\tau_{A_3}^r(1)$	—	$\tau_s^{A_3}(1)$	N
...	...	...	...	...	...	...

We now consider the case that when node A_1 attacks the synchronization process by illegitimately claiming the identity of node r, nodes m and s would receive the incorrect timestamps $\tau_{A_1}^r(k)$ simultaneously. Since the difference of received two consecutive timestamps with identity r exceeds the preset threshold, existing threshold detection-based secure synchronization protocols would regard node r as a malicious node. Therefore, secure node r would be isolated by the other nodes. On the contrary, our RSTS defense mechanism does not detect whether the received messages are valid. Note that the purpose of RSTS is to obtain a valid $\Delta\tau_{ms}(t) = \tau_m(t) - \tau_s(t)$, where the records $\tau_m(t)$ and $\tau_s(t)$ are based on the same basic time. Although the sending timestamp $\tau_{A_1}^r(k)$ from the attacker A_1 is incorrect, the receiving timestamps of nodes m and s refer to the same reference and the difference $\tau_m^{A_1}(k) - \tau_s^{A_1}(k)$ is correct. Hence, the invalid messages can

be filtered out automatically by the time difference when receiving the public timestamps from the attacker.

If the attacker A_2 advertises invalid timestamps $\tau^r_{A_2}(k)$ with disguised identity r, only node m could receive the timestamps. Based on the matching principle of the received message sequence number, there are no corresponding messages $\tau_s^{A_2}(k)$ in node s. Consequently, the destructiveness from attacker A_2 is automatically relieved. Similarly, for attacker A_3, the threat no longer exists by default.

4.2. Multi-Hop Network Synchronization

RSTS focuses primarily on layered network topology. Although the global synchronization methods in RBS and FTSP are available, RSTS works in fundamentally different ways, which results in significant notes in constructing a multi-hop synchronization path.

For example, Figure 4a shows an example of a multi-hop chain topology in FTSP, and Figure 4b shows the corresponding synchronization logical graphs of RSTS, where the black arrow represents the communication relationship between two nodes and the green arrow is the synchronization path. Node 0 is the root, also known as the time source, and other ordinary nodes synchronize with node 0 hop-by-hop. It can be observed that the black arrows are unidirectional and bidirectional in Figure 4a,b, respectively, which indicates that the transmission direction of time information is not the same in FTSP and RSTS. Each ordinary node can work as a master or a slave in both FTSP and RSTS. Moreover, RSTS's node has an additional role, the reference node, which serves as a shared neighbor node. For each node, it can work as a master after achieving synchronization with the upper node. To ensure the composition of the global network time synchronization path, at least one triangle sub-topology must exist in the network, as shown in Figure 4b, 0↔1↔10↔0.

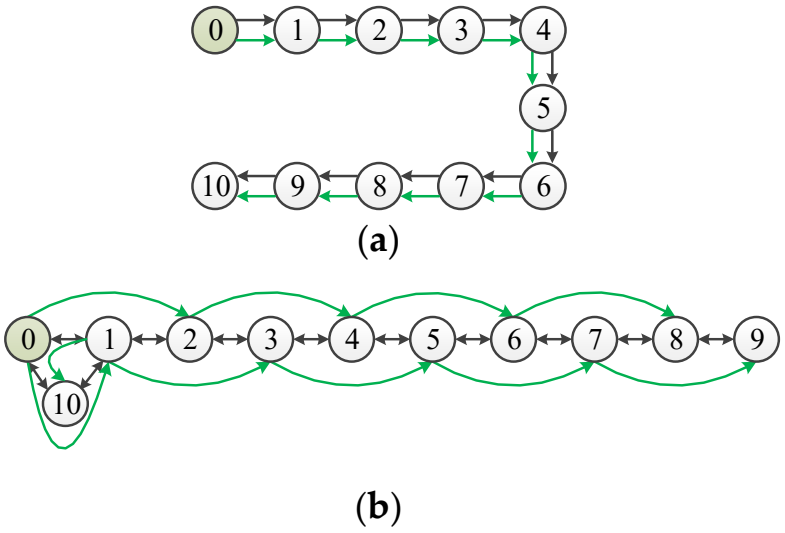

Figure 4. Multi-hop chain topology. (**a**) FTSP; (**b**) RSTS.

In the comparison of the synchronization path, we can see that RSTS has a potential advantage in shortening the longest synchronization path of the network, i.e., 10 in FTSP and 5 in RSTS as shown in Figure 4, and the closer the triangle sub-topology is to the root node, the better the effect. Meanwhile, in layered topology, the node broadcasts synchronization messages to other nodes in the lower level only after entering the sync state. The synchronization accuracy will decrease as the synchronization path increases and the distant nodes may wait a long time to synchronize. Hence, RSTS shows a better performance than FTSP in improving the network synchronization accuracy and convergence speed.

4.3. Security Analysis

FTSP is designed for a benign environment in which each node is valid. However, malicious attacks would destruct the synchronization process by injecting irregular time information or masquerading as a normal node. Therefore, in this subsection, we analyze the security of FTSP, FTSP with threshold detection, and RSTS.

Under manipulation attacks. The manipulation attacker injects illegal timestamps into the network misleading its neighbor to synchronize to the wrong time. Obviously, unprotected FTSP quickly loses synchronization. Since the timestamps from attackers are without rules and two timestamps from a given valid neighbor are said to be conforming [21], many secure methods design checking mechanisms based on a preset threshold, such as delay and the difference of relative clock skew. If a node's timestamps cannot pass the threshold filtering, the node may be blacklisted. The network would exclude the suspicious node from the synchronous path and rebuild the topology. The feasibility of these methods has been proven. In RSTS, the attacker can act as a reference node or master node broadcasting illegal time. RSTS's security has been analyzed in the above illustration when the attacker acts as a reference node. The other scenario is similar to FTSP with threshold detection. Hence, it can integrate threshold detection to enhance RSTS's defense capability.

Under Sybil attacks. Although the threshold detecting mechanisms are available in defending against manipulation attacks, they may lose defensive ability in facing Sybil attacks. Unlike manipulation attacks, a Sybil attacker masquerades as a normal node to broadcast illegal information instead of broadcasting illegal information directly. In this case, threshold-detecting mechanisms may render the disguised normal node out of synchronization. RSTS adopts a novel forwarding-based synchronization path and filters the attacker's time information by default in estimating the synchronization parameters, which results in a secure synchronization under Sybil attacks.

4.4. Communication Energy Cost

RSTS improves synchronization security by making use of the public neighbor forwarding mechanism. This protocol gains an additional accuracy over FTSP due to shortening the synchronization path by nearly half. Unfortunately, the slave node cannot obtain the master's timestamps efficiently, which results in higher communication energy costs. We assume that each broadcast costs energy E. In FTSP, each node updates its clock with n communications and costs energy nE. For RSTS, it adds an information forwarding step with two communications, which results in energy $3nE$. In other words, RSTS secures time synchronization at the cost of increasing the communication load, which is a common issue for secure time synchronization protocols.

5. Evaluation

To verify RSTS's performance, we conducted simulations in Matlab R2021b and implemented a 10-hop chain topology with 11 nodes, as shown in Figure 4. For comparison, we also performed FTSP with or without threshold detection, i.e., the difference between two consecutively receiving timestamps did not exceed the preset threshold of 1.001. The clock skew α_i and offset β_i of each node were randomly selected from the sets [0.9999, 1.0001] and [0, 0.0002], respectively, as the maximum clock drift was 100 ppm [9]. The synchronization period T was 1 s. The length of the linear regression table was 8 [23]. Each node recorded its time based on a 32.768 kHz crystal oscillator. We also defined the clock difference between each ordinary node and the time source to denote the synchronization performance, including clock skew and offset. To simulate a real network environment, we assumed that the communication delay satisfies a normal distribution with a mean of 2.5×10^{-4} and a variance of 1×10^{-8} [9].

5.1. Under No Attacks

Figure 5 compares the performance of FTSP and RSTS in a benign environment, i.e., the difference of clock skew and offset, and the error bars of clock offset that indicate the estimation error average and standard deviation. We can see that both FTSP and RSTS eventually achieve synchronization. Since FTSP adopts a hop-by-hop synchronization manner, it can be observed that as the hop gets larger, the estimated synchronization error between ordinary nodes and the time source gradually increases. Figure 5b shows that under the same network scale, the synchronization accuracy of RSTS is significantly higher than that of FTSP, as the public neighbor forwarding mechanism shortens the synchronization path between the farther node and the time source. To further explain, in FTSP, the farther node 9 synchronizes with source node 0 after 9 hops, while its synchronization path is just 5 in RSTS, and its synchronization accuracy is almost improved by one order of magnitude.

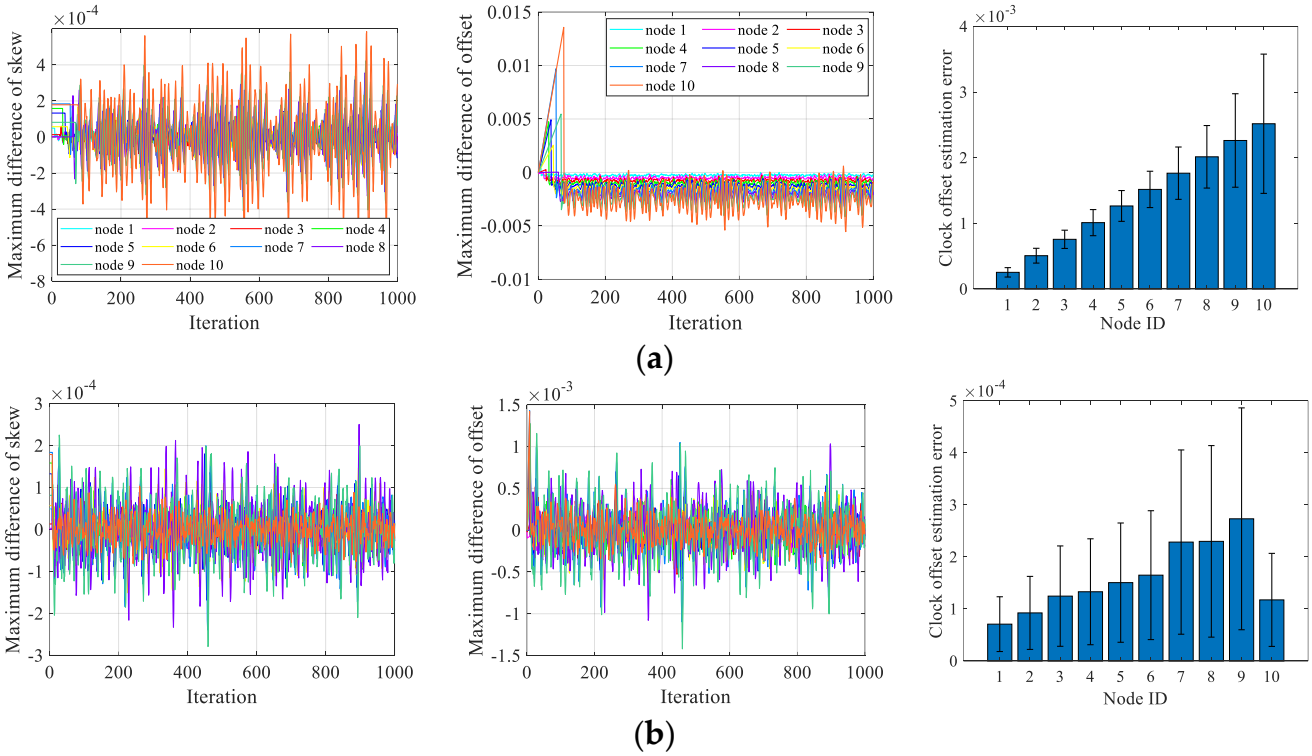

Figure 5. Performance of FTSP and RSTS under no attacks. The representation of the lines is the same in (**a**,**b**). (**a**) FTSP; (**b**) RSTS.

5.2. Under Manipulation Attacks

We proceed to compare the performance of FTSP, FTSP with threshold detection, and RSTS under manipulation attacks. Assume that nodes 3 and 6 in the chain topology are manipulation attackers who broadcast invalid timestamps every $3T$ periods and attack the network after 100 iterations. The attack power ω is randomly selected in [0, 0.01] s. With the existence of manipulation attackers, the time source's time information cannot be effectively transmitted to the nodes after node 2. The synchronization performance of safe nodes is shown in Figure 6. We can see that safe nodes 1 and 2 can still synchronize with the time source, but the subsequent nodes lose synchronization in Figure 6a. For threshold detection, invalid timestamps could not pass the detection process. Hence, malicious nodes are marked and excluded from the network. RSTS updates the clock based on the difference in receiving timestamps and removes attackers' unfavorable effects. Comparing Figure 5a with Figure 6, we can see that both RSTS and threshold detection can defend against manipulation attacks. Moreover, RSTS exhibits a higher synchronization accuracy.

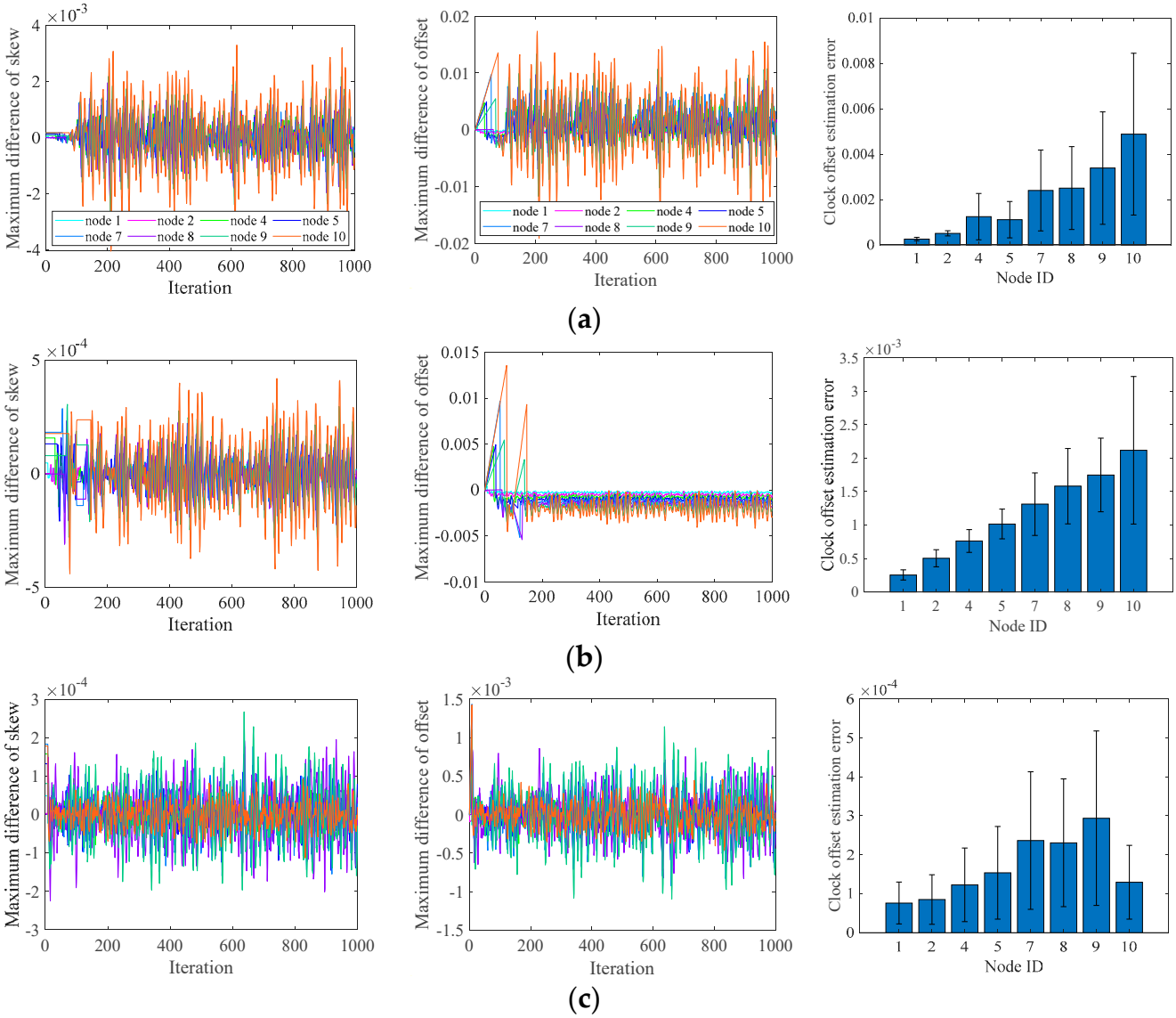

Figure 6. Performance of FTSP, FTSP with threshold detection, and RSTS under manipulation attacks. The representation of the lines is the same in (**a**–**c**). (**a**) FTSP; (**b**) FTSP with threshold detection; and (**c**) RSTS.

5.3. Under Sybil Attacks

In this section, we study the performance of these defense mechanisms under Sybil attacks. Assume that the attacker can randomly present as nodes 3 and 6. It broadcasts invalid timestamps every 3T periods. In Figure 7a, clearly, FTSP cannot converge under a Sybil attack. Since timestamps with ID 3 or 6 include valid and erroneous time information, these timestamps are more irregular in comparison with manipulation attacks. Nodes 4 and 5 may synchronize to a wrong clock, as could nodes 7, 8, 9, and 10. The safe nodes 3 and 6 disguised by the attacker would be mistaken for malicious nodes in threshold detection and be broken up with the network, as shown in Figure 7b. In contrast, Figure 7c shows that RSTS is robust against a Sybil attack since it does not disrupt the communication with the suspicious nodes and exploits the correct difference of receiving timestamps for synchronization.

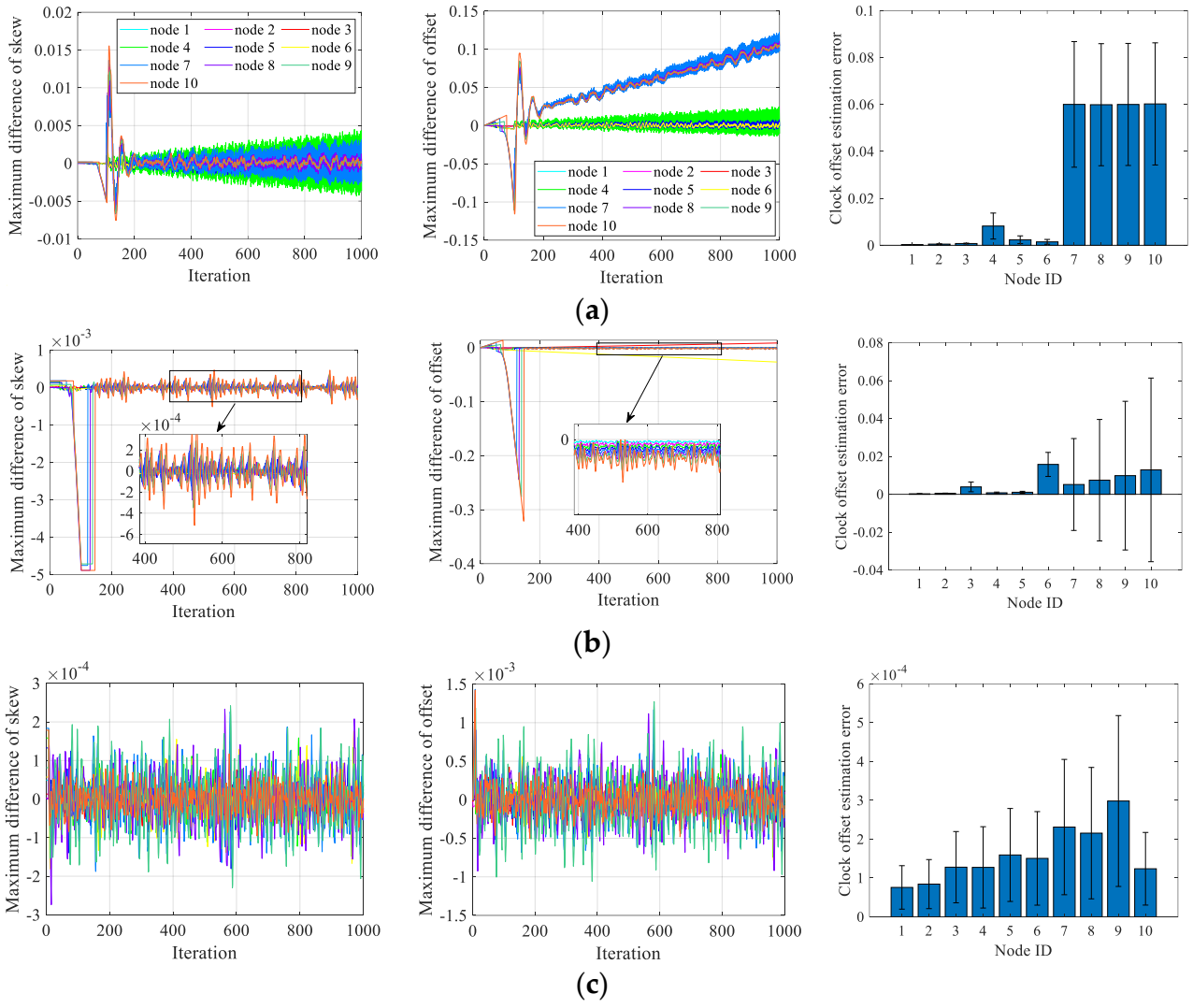

Figure 7. Performance of FTSP, FTSP with threshold detection, and RSTS under Sybil attacks. The representation of the lines is the same in (**a**–**c**). (**a**) FTSP; (**b**) FTSP with threshold detection; and (**c**) RSTS.

6. Conclusions

In this paper, we propose a novel secure time synchronization protocol against Sybil attacks for the centralized network employing a public neighbor forwarding mechanism based on reference broadcast, i.e., RSTS. The simulation results are given to show that RSTS is valid against Sybil attacks as well as manipulation attacks. Specifically, RSTS invalidates the potential malicious attacks at the message level instead of isolating the suspicious nodes. Moreover, using non-adjacent two-hop neighbors as the master nodes greatly shortens the synchronization path and thus invisibly improves the network synchronization accuracy.

Although analysis and simulation results have proven RSTS's security, there are still multiple future research directions. First, it should be noted that RSTS secures the network synchronization at the cost of improving communication overhead, we would like to balance the communication overhead and security. Second, it would be interesting to devise a more complex tree topology constructing method with RSTS and expand RSTS's core idea to existing distributed synchronization protocols.

Author Contributions: Conceptualization, Z.W. and C.Y.; methodology, Z.W. and D.S.; software, D.S. and C.Y.; validation, Z.W., D.S. and C.Y.; formal analysis, Z.W., D.S. and C.Y.; investigation, Z.W. and D.S.; resources, Z.W.; data curation, D.S.; writing—original draft preparation, Z.W., D.S. and C.Y.; writing—review and editing, Z.W. and C.Y.; visualization, D.S.; supervision, Z.W.; project administration, Z.W.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 62002140), the Natural Science Foundation of Jiangsu Province (Grant No. BK20200887), and the Doctor's Program of Entrepreneurship and Innovation in Jiangsu Province.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, S.A.A.; Ovsthus, K.; Kristensen, L.M. An industrial perspective on wireless sensor networks—A survey of requirements, protocols, and challenges. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1391–1412.
2. Vitturi, S.; Zunino, C.; Sauter, T. Industrial communication systems and their future challenges: Next-generation Ethernet, IIoT, and 5G. *Proc. IEEE* **2019**, *107*, 944–961. [CrossRef]
3. Koo, Y.C.; Mahyuddin, M.N.; Wahab, M.N.A. Novel control theoretic consensus-based time synchronization algorithm for WSN in industrial applications: Convergence analysis and performance characterization. *IEEE Sens. J.* **2023**, *23*, 4159–4175.
4. Zong, Y.; Dai, X.; Wei, Z.; Zou, M.; Guo, W.; Gao, Z. Robust time synchronization for industrial internet of things by H∞ output feedback control. *IEEE Internet Things J.* **2023**, *10*, 2021–2030. [CrossRef]
5. Lo Bello, L.; Steiner, W. A perspective on IEEE Time-Sensitive Networking for industrial communication and automation systems. *Proc. IEEE* **2019**, *107*, 1094–1120.
6. Kerö, N.; Puhm, A.; Kernen, T.; Mroczkowski, A. Performance and reliability aspects of clock synchronization techniques for industrial automation. *Proc. IEEE* **2019**, *107*, 1011–1026.
7. Shi, F.; Yang, S.X.; Mukherjee, M.; Jiang, H.; da Costa, D.B.; Wong, W.K. Parameter-sharing-based average-consensus time synchronization in IoT networks. *IEEE Internet Things J.* **2023**, *10*, 8215–8227. [CrossRef]
8. Huan, X.; He, H.; Wang, T.; Wu, Q.; Hu, H. A timestamp-free time synchronization scheme based on reverse asymmetric framework for practical *resource*-constrained wireless sensor networks. *IEEE Trans. Commun.* **2022**, *70*, 6109–6121. [CrossRef]
9. He, J.; Cheng, P.; Shi, L.; Chen, J.; Sun, Y. Time synchronization in WSNs: A maximum-value-based consensus approach. *IEEE Trans. Autom. Control* **2014**, *59*, 660–675.
10. Schenato, L.; Fiorentin, F. Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica* **2011**, *47*, 1878–1886.
11. Ye, K.; Yan, Y.; Wu, H. Time synchronization algorithm for networked control systems based on stochastic search. *IEEE Trans. Ind. Inform.* **2022**, *18*, 26–34. [CrossRef]
12. Cao, B.; Zhao, J.; Gu, Y.; Fan, S.; Yang, P. Security-aware industrial wireless sensor network deployment optimization. *IEEE Trans. Ind. Inform.* **2020**, *16*, 5309–5316. [CrossRef]
13. Angueira, P.; Val, I.; Montalban, J.; Seijo, O.; Iradier, E.; Fontaneda, P.S.; Fanari, L.; Arriola, A. A survey of physical layer techniques for secure wireless communications in industry. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 810–838. [CrossRef]
14. Weng, Y.; Zhang, Y. A survey of secure time synchronization. *Appl. Sci.* **2023**, *13*, 3923. [CrossRef]
15. He, J.; Cheng, P.; Shi, L.; Chen, J. SATS: Secure average-consensus-based time synchronization in wireless sensor networks. *IEEE Trans. Signal Process.* **2013**, *61*, 6387–6400.
16. He, J.; Chen, J.; Cheng, P.; Cao, X. Secure time synchronization in wireless sensor networks: A maximum consensus-based approach. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 1055–1065.
17. Du, X.; Guizani, M.; Xiao, Y.; Chen, H.H. Secure and efficient time synchronization in heterogeneous sensor networks. *IEEE Trans. Veh. Technol.* **2008**, *57*, 2387–2394.
18. Rahman, M.; El-Khatib, K. Secure time synchronization for wireless sensor networks based on bilinear pairing functions. *IEEE Trans. Parallel Distrib. Syst.* **2010**. [CrossRef]
19. Qiu, T.; Liu, X.; Han, M.; Ning, H.; Wu, D.O. A secure time synchronization protocol against fake timestamps for large-scale internet of things. *IEEE Internet Things J.* **2017**, *4*, 1879–1889. [CrossRef]
20. Jia, P.; Wang, X.; Zheng, K. Distributed clock synchronization based on intelligent clustering in local area industrial IoT systems. *IEEE Trans. Ind. Inform.* **2020**, *16*, 3697–3707. [CrossRef]
21. Dong, W.; Liu, X. Robust and secure time-synchronization against sybil attacks for sensor networks. *IEEE Trans. Ind. Inform.* **2015**, *11*, 1482–1491. [CrossRef]

22. Wang, Z.; Zeng, P.; Kong, L.; Li, D.; Jin, X. Node-identification-based secure time synchronization in industrial wireless sensor networks. *Sensors* **2018**, *18*, 2718. [PubMed]
23. Maroti, M.; Kusy, B.; Simon, G.; Ledeczi, A. The flooding time synchronization protocol. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys), Baltimore, MD, USA, 3–5 November 2004; pp. 39–49.
24. Elson, J.; Girod, L.; Estrin, D. Fine-grained network time synchronization using reference broadcasts. In Proceedings of the 5th USENIX Symposium on Operating Systems Design and Implementation, Boston, MA, USA, 9–11 December 2002; pp. 147–163.
25. He, J.; Duan, X.; Cheng, P.; Shi, L.; Cai, L. Accurate clock synchronization in wireless sensor networks with bounded noise. *Automatica* **2017**, *81*, 350–358. [CrossRef]
26. Gong, F.; Sichitiu, M.L. CESP: A low-power high-accuracy time synchronization protocol. *IEEE Trans. Veh. Technol.* **2016**, *65*, 2387–2396. [CrossRef]
27. Huan, X.; Kim, K.S.; Lee, S.; Lim, E.G.; Marshall, A. A beaconless asymmetric energy-efficient time synchronization scheme for resource-constrained multi-hop wireless sensor networks. *IEEE Trans. Commun.* **2020**, *68*, 1716–1730. [CrossRef]
28. Moussa, B.; Kassouf, M.; Hadjidj, R.; Debbabi, M.; Assi, C. An extension to the Precision Time Protocol (PTP) to enable the detection of cyber attacks. *IEEE Trans. Ind. Inform.* **2019**, *16*, 18–27. [CrossRef]
29. Sun, K.; Ning, P.; Wang, C. Secure and resilient clock synchronization in wireless sensor networks. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 395–408.
30. Phan, L.A.; Kim, T.; Kim, T. Robust neighbor-aware time synchronization protocol for wireless sensor network in dynamic and hostile environments. *IEEE Internet Things J.* **2021**, *8*, 1934–1945. [CrossRef]
31. Kumar, D.U.S.R.; Rajamani, V. Detecting and preventing sybil attacks in wireless sensor networks using message authentication and passing method. *Sci. World J.* **2015**, *2015*, 841267.
32. Balachandran, N.; Sanyal, S. A review of techniques to mitigate sybil attacks. *arXiv* **2012**, arXiv:1207.2617. [CrossRef]