IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 30, 2020, accepted June 11, 2020, date of publication June 15, 2020, date of current version June 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3002577

Finite-Time Consensus-Based Clock Synchronization Under Deception Attacks

YIMING WU[1] **AND XIONGXIONG HE**[2]
[1]School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
[2]College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

Corresponding author: Yiming Wu (ymwu@hdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61803135 and Grant 61873239, in part by the Cyberspace Security Major Program of the National Key Research and Development Plan of China under Grant 2016YFB0800201, and in part by the Zhejiang Provincial Public Welfare Research Project of China under Grant LGG18F020015.

ABSTRACT This paper concentrates on the finite-time clock synchronization problem for wireless sensor networks (WSNs) under deception attacks. Compared with adding additional communication links to a network, we introduce a new mechanism termed as ‘‘trusted link’’ to improve the resilience of the network, and show that with small changes (set a fraction of links as the trusted links) in the network structure the network robustness for deception attacks can be improved significantly. Then, an iterative learning control based consensus control methodology with built-in attack mitigation mechanism is proposed. Not only the security and robustness are guaranteed by the proposed controller, but also the convergence time is fixed, which makes the synchronization algorithm more suitable for practical WSNs. Finally, simulation results are provided to demonstrate the effectiveness of the theoretical results.

INDEX TERMS Wireless sensor networks, consensus, robustness, finite-time, clock synchronization.

I. INTRODUCTION

The research on wireless sensor networks (WSNs) has gained a lot of attention over the last few decades because of their wide applications in various areas, such as military, environmental, medical, and industrial domains. Accurate clock or time synchronization plays a fundamental role in studying WSNs since various applications such as mobile object tracking [1], data fusion [2], and public infrastructure surveillance [3] are requiring that all sensor nodes have a common time reference. Up to now, a number of clock synchronization algorithms in various scenarios have been proposed [4]–[7].

Recently, consensus-based clock synchronization which in a fully distributed scheme has attracted increasing research attention [6], [8]–[10]. It is developed to overcome the shortages of traditional root-based or tree-based clock synchronization protocols in terms of increasing the scalability and robustness of synchronization. However, most existing works all assume the system is deployed in a benign environment in which every sensor node of the networks is fault-free and the information channel between nodes are reliable, and very few of the previous schemes has been designed with necessary security measures in mind. Traditional consensus-based synchronization schemes are quite vulnerable to different types of attacks. Even when a single node in a WSN is compromised by the attacker and starts to exchange false information with its neighbors, this will eventually lead to invalid whole consensus process [11]. Therefore, investigations on consensus-based clock synchronization for WSNs against malicious attacks are desirable.

Because of the limited computation and communication capabilities of each sensor node in WSNs, existing security methods such as the cryptographic techniques and attack detection and identification techniques are usually difficult to apply. Without identifying misbehaving nodes, LeBlanc and Koutsoukos in [12] present a resilient asymptotic weakly stable synchronization protocol in time-varying network, and ensure synchronization can be achieved in the presence of up to F malicious adversaries. The paper [13] studies secure consensus in synchronous networks under message manipulation attacks, and proposed a secure synchronous consensus algorithm based on two-hop neighboring

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

information in the network. To avoid performance deterioration in distributed networks, Kailkhura *et al.* in [14] propose a weighted average consensus algorithm that is robust to data falsification attacks.

However, as explained in aforementioned works [12]–[14], only in a sufficiently high network robustness, consensus-based synchronous can be achieved under attacks. In order to improve network robustness, a conventional way is achieved by adding further communication links between nodes, i.e., by increasing redundancy. However, it may be prohibitively expensive or impossible in practice. More recently, a novel idea for increasing structural robustness without adding extra links is proposed in [15], where the basic strategy is to make a small subset of nodes trusted, that is, immune to attacks. With the help of trusted nodes, our earlier work [16] discussed the secure consensus problem for the first-order and second-order heterogeneous system. And Mitra *et al.* [17] also used the trusted nodes to address the issue of distributed state estimation of a linear dynamical process in an attack-prone environment.

Besides, achieving convergence in a finite time is another important desirable property for consensus based clock synchronization problem, while little research has addressed this topic in the context of the network under attacks. In practical situation, finite-time consensus-based algorithm ensures clock synchronization within a limited time interval meanwhile with computation and communication cost reduced, thus enabling better application in WSNs. In addition, as claimed in [18], the networks with finite-time consensus convergence usually have better performance in the disturbance rejection and robustness against uncertainties. A distributed finite-time consensus protocol which achieves agreement with respect to the median value of the initial states and is robust to the influence of uncooperative nodes was proposed in [19]. Different from many previous works on finite time consensus networks with all cooperative interactions, Meng *et al.* [20] first studied the finite-time consensus problems on networks in the presence of antagonistic interactions. The authors in [21] studied the attack tolerant finite-time consensus problems for continuous-time multi-agent networks under directed topologies. The authors in [22] studied the both continuous time and discrete time systems in the presence of misbehaving agents, and proposed a norm-based filtering mechanism which guarantees convergence in finite-time even with bounded inputs. However, it is important to remark that the convergence rates of consensus-based synchronization algorithms proposed by [18]–[22] can be influenced by the network connectivity, i.e., the second-smallest eigenvalue of the interaction graph Laplacian matrix. Therefore, these synchronous algorithm implementations might be very sensitive to node and link failures.

Recently, iterative learning control (ILC) is regarded as an effective control strategy, which can give an alternate solution to solve the finite-time consensus problem. Different from existing works [18]–[22], the ILC-based consensus algorithm which fully utilizes the past control experience to improve the performance of consensus processes in the current iteration, not only has better robust to the connectivity of the interconnection topology, but also has prescribed terminal time as desired [23]. Owing to its simplicity and effectiveness, ILC-based consensus has generated considerable interest over the past years [24]–[28].

Motivated by these, in this paper, we concern with the finite-time consensus problem for networks under adversarial attacks. To improve the resilience of networks against deception attacks, we first introduce and analyze the notion of r-robustness with the help of the trusted link mechanism. It is shown that the robustness of a network to tolerate deception attacks can be effectively improved by setting a small subset of links as the trusted links. Then, we develop a distributed finite-time clock synchronization protocol by adopting the concept of ILC-based consensus which provides precise converging time under deception attacks in WSNs.

The major contributions of this work are summarized as follows:

1) We propose and characterize the notion of network robustness based on the concept of trusted links inspired by [15], and show that network connectivity can be significantly improved without adding the additional links.
2) By exploiting the principle of r-robustness with trusted links, an ILC-MSR based consensus algorithm is proposed in this paper. It is shown that all nodes can be guaranteed to achieve consensus in a finite time with the proposed protocol.
3) A novel clock synchronization based on the proposed consensus algorithm for WSNs is designed, and experimental results are presented to demonstrate the effectiveness of the proposed protocol and design method.

The rest of the paper is organized as follows: We review some knowledge of graph theory and attack model, and formulate the problem in Section II. In Section III, necessary and sufficient (algebraic and graphical) conditions are analyzed for ILC-based consensus problem under deception attacks. An application in the clock synchronization for WSNs is presented in Section IV to validate the effectiveness of the proposed control strategy. The simulation results and conclusion will be showed in section V and VI, respectively.

Notations: Throughout the paper, the symbols $\mathbb{R}$, $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$ represent the set of real numbers, n-dimensional real vectors, and $m \times n$ real matrices, respectively. Denote by $\mathbf{1}_n$ the n-dimensional vector of ones and denote by I_n the $n \times n$ identity matrix. For any matrix, $A \in \mathbb{R}^{n\times n}$, $A > 0$ denotes its positive definite, $A \geq 0$ denotes its positive semi-definite, and $(\cdot)^{\mathrm{T}}$ denotes the transpose.

II. PRELIMINARIES AND PROBLEM FORMULATION

In this section, first some basic concepts in the graph theory and attack model that will be used throughout the paper are reviewed, then the problem to be considered is formulated.

A. ALGEBRAIC GRAPH THEORY

A weighted directed graph (digraph for short) is represented as a triple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is a non-empty set of nodes, $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ is a set of edges, and $A \in \mathbb{R}^{n\times n}$ is a weighted adjacency matrix. An edge of $\mathcal{G}$ is denoted by $e_{ij} = (j, i)$, where the first element i of e_{ij} is said to be the head of the edge and the other j to be the tail. If $(j, i) \in \mathcal{E}$, node j is called an in-neighbor of i. Then, the set of i' in-neighbors is denoted by $\mathcal{N}_i = \{j \mid (j, i) \in \mathcal{E}\}$, and the set of incoming edges of i is denoted by $\mathcal{E}_i = \{(j, i) \mid j \in \mathcal{N}_i\}$. $a_{ij} \in A$ is called the weight of edge (j, i), and $a_{ij} \in [\mu, 1)$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise, where $\mu \in (0, 1)$. Moreover, the self-loop is not considered in this paper, i.e., $(i, i) \notin \mathcal{E}$, $\forall i \in \mathcal{V}$. A directed path is a sequence of ordered edges of the form $(i_1, i_2), (i_2, i_3), \ldots$, where $i_j \in \mathcal{V}$. A digraph is said to have a spanning tree if there exists at least one node i, such that for any other node j there is a path from i to j.

Next, we introduce several concepts of network robustness in digraphs, which introduced in [29], and later studied in [30] and [31].

Definition 1: (*r-reachable*): A nonempty set $\mathcal{S} \subseteq \mathcal{V}$ is said to be r-reachable if there is at least one node $i \in \mathcal{S}$ such that $|\mathcal{E}'_i| \geq r$, where $\mathcal{E}'_i = \{(j, i) \in \mathcal{E} : j \in \mathcal{N}_i \backslash \mathcal{S}\}$ denotes the set of i' incoming edges from nodes outside of $\mathcal{S}$.

Definition 2: (*r-robust*): A digraph is said to be r-robust if for every pair of nonempty, disjoint subsets of $\mathcal{V}$, at least one of the subsets is r-reachable set.

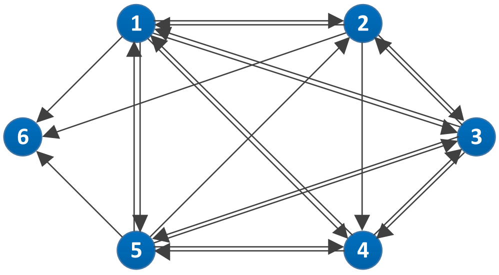

FIGURE 1. **Network topology satisfies 3-robust.**

In Fig. 1, we display an example graph with six nodes. It has just enough connectivity to be 3-robust. If any of the links are removed, one can check that this level of network robustness will be lost.

From Definitions 1 and 2, it is easy to get the following lemmas.

Lemma 1: A digraph $\mathcal{G}$ is s-robust, where $1 \leq s < r$, if $\mathcal{G}$ is a r-robust graph.

Lemma 2: If $\mathcal{G}$ is an r-robust graph, then after removing up to s incoming links of each node in $\mathcal{G}$, where $0 \leq s < r$, the remaining graph is an $(r\text{-}s)$-robust graph.

B. ATTACK MODEL

Among various cyber-attacks, deception attacks, which can also be called as false data injection attacks, are of high risk and can cause cascading effects on distributed systems. In a deception attack, the adversaries can modify the data packets being transmitted in wireless communication channels, and floods the network with false-data by taking full advantage of distributed protocol. In this paper, we assume that the deception attack means that in-neighbouring information of nodes are compromised and modified. In other words, a communication link (i, j) is said to be compromised by the deception attack if the message sent by node i is different from the message received by node j in the time iteration. Considering the fact that the adversary has limited capability in practice (i.e., the adversary does not have the complete capability to compromise all the communication links of the underlying networks), a widely adopted attack model is so called "F-local model" in the previous literature on multi-agent consensus problems, for example, [12], [16], [22], [32]. In this work, we also consider that there exists an upper bound F on the number of compromised links of each node's incoming links. The detailed assumption of our attack model is as follows.

Assumption 1: (*F-local deception attack*) Given a digraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, for any node $i \in \mathcal{V}$, there are at most F compromised links within i' whole incoming links.

C. PROBLEM FORMULATION

Consider a distributed network consisting of n nodes, the communication topology is a weighted digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$. At the k-th iteration, the dynamics of the i-th node take the following form:

$$\dot{x}_i^k(t) = u_i^k, \quad t \in [0, T], \tag{1}$$

where $k = 1, 2, \ldots$ label different iterations, while $T > 0$ represents the operation time in each iteration. $x_i^k(t) \in \mathbb{R}$ is the state of the i-th node, and $u_i^k \in \mathbb{R}$ is the control input or protocol for i to be designed.

Since it is required that the systems could perform the same consensus task repeatedly, the initial state needs to be reset at each iteration. A commonly reset condition which is adopted in many existing works [27], [33] is that just simply reset to the same value, i.e.,

$$x_i^k(0) = x_{i0}, \quad k = 1, 2, \ldots \tag{2}$$

where x_{i0}, $i \in \mathcal{V}$ is the initial state value of the system.

To be able to address the above problems, we first state the definition of finite-time consensus under deception attacks. Let $x_{\min}^k$ and $x_{\max}^k$ be the minimum and maximum state value for all $i \in \mathcal{V}$ in time interval $[0, T]$, respectively. Then we provide the following definition.

Definition 3: We say a system reaches finite-time consensus under F-local deception attack, if the system can satisfy the following conditions:

(i) For any time t, the state value of each node $x_i^k(t)$, $i \in \mathcal{V}$ is always in the interval $[x_{\min}^0, x_{\max}^0]$;

(ii) There exists a finite time T, such that $\lim_{k\to\infty}(x_j^k(T) - x_i^k(T)) = 0$, $\forall i, j \in \mathcal{V}$;

where $x_{\min}^0$ and $x_{\max}^0$ denote the minimum and maximum initial state values of all nodes, respectively.

The first condition is typically referred to as the *safety* or *validity* condition, which means that all nodes' values in the network must stay in the convex hull of initial values during the whole consensus process. It is important in some certain safety critical applications, whenever $[x^0_{\min}, x^0_{\max}]$ is a known safe set. The second condition is a finite-time convergence condition on agreement.

Therefore, generally speaking, our goal in this paper is to design a distributed controller to achieve the consensus in the presence of deception attacks and meanwhile ensure the convergence within a given time interval.

III. FINITE TIME CONSENSUS UNDER DECEPTION ATTACKS

A. NETWORK ROBUSTNESS WITH TRUSTED LINKS

In this subsection, we will explore a necessary and sufficient graphical condition such that our control protocol to be designed later can be effective in the presence of deception attacks.

First, we introduce a new concept, called *trusted link*. We assume the message can flow correctly through the paths formed by trusted links. That is, insusceptible to message manipulation attacks. The formal definition is as follows.

Definition 4: (*trusted link*): We say a communication link in $\mathcal{G}$ is a trusted link, if all message conveyed on it cannot be wiretapped, modified or failed by the deception attacks.

Since the structure of network topology considered in [30]–[32] is not involved in the trusted links, we need to modify the previous definition of the network robustness. Let $\mathcal{E}_\mathcal{T}$ and $\mathcal{E}_\mathcal{A}$ be the trusted link set and the compromised link set, respectively. Considering the existence of trusted links, we provide the redefinition of network robustness, as described below.

Definition 5: (*r-reachable with* $\mathcal{E}_\mathcal{T}$): A nonempty set $\mathcal{S} \subseteq \mathcal{V}$ is said to be r-reachable with $\mathcal{E}_\mathcal{T}$ if there is at least one node $i \in \mathcal{S}$ such that $|\mathcal{E}'_i| \geq r$ or $\mathcal{E}'_i \cap \mathcal{E}_\mathcal{T} \neq \varnothing$, where $r \in Z_{\geqslant 0}$.

Definition 6: (*r-robust with* $\mathcal{E}_\mathcal{T}$): A digraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is said to be r-robust with $\mathcal{E}_\mathcal{T}$ if for every pair of nonempty, disjoint subsets of $\mathcal{V}$, at least one of the subsets is r-reachable with $\mathcal{E}_\mathcal{T}$.

The main idea of the network robustness is to provide insights about purely local diffusion dynamics over distributed networks so as to ensure that each pair of disjoint and nonempty subsets of nodes in the network can receive enough messages from the nodes outside of one's own set. In other words, it guarantees the information flow in a distributed network.

Remark 1: Notice that to obtain a specific r-robust network, the given redefinition of network robustness does not require communication links as much as earlier definition in [29]–[31], which may be desirable to reduce the communication burden of the system. On the other hand, a network with higher network robustness may easier to constitute if trusted communication channels can be established between a small portion of nodes.

Lemma 3: For a digraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with a set of trusted links $\mathcal{E}_\mathcal{T}$, $\mathcal{G}$ is *infinity*-robust if $\mathcal{V}$ and a subset of $\mathcal{E}_\mathcal{T}$ can form a spanning tree.

Proof of Lemma 3: We assume that the robustness of $\mathcal{G}$ cannot reach infinity. Then, according to Definition 6, we know that there must exist a pair of disjoint and nonempty subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$, neither $\mathcal{S}_1$ nor $\mathcal{S}_2$ is an infinity-reachable set, which means all nodes in $\mathcal{S}_1$ and $\mathcal{S}_2$ do not has an incoming trusted link from outside of one' own set. It contradicts to the definition of a spanning tree. Therefore, $\mathcal{G}$ is *infinity*-robust. ■

Lemma 4: For a digraph $\mathcal{G}$ with a set of trusted links $\mathcal{E}_\mathcal{T}$, $\mathcal{G}$ contains a spanning tree if $\mathcal{G}$ is 1-robust with $\mathcal{E}_\mathcal{T}$.

Proof of Lemma 4: We prove this by contradiction. Assume that $\mathcal{G}$ does not contain a spanning tree. Let A be the adjacency matrix of $\mathcal{G}$. Then according to Seneta [34], it has that matrix A is decomposable, which means one can split the graph $\mathcal{G}$ into two disjoint subsets $\mathcal{S}_1$ and $\mathcal{S}_2$, and no information exchange happens between them, which contradicts the definition of 1-robust with $\mathcal{E}_\mathcal{T}$. ■

Remark 2: Since robustness of a network can be improved by setting a set of trusted links, one question is that how to quantify the impact of setting the trusted links on the robustness of an arbitrary network. To solve this problem, a prerequisite is that one could accurately determine the robustness of the network before and after adding the trusted links. Though the resilient consensus (or synchronization) algorithms based on the network robustness have been widely studied, how to determine the robustness of a given network is still a challenging problem. One of the biggest difficulties in this issue is that the problem of determining the r-robustness of a network is NP-hard [30]. Therefore, finding efficient ways of determining the robustness of arbitrary graphs in general remains an open problem, which can be our future work.

Remark 3: While being used for improving the robustness of a network, the advantages of trusted links can be a reduction in the total number of communication links. However, we find that finding a minimum set of trusted links that achieve certain network robustness is a computationally hard problem, which may require further investigation.

B. DESIGN OF CONTROL LAW

Next, we come to design our control input u_i^k when the digraph $\mathcal{G}$ satisfies some specific network robustness. The algorithm will be referred to as the Iterative Learning Control-Based Mean Subsequence Reduced (ILC-MSR) algorithm, which actually is an iterative version of the Weighted-MSR algorithm proposed in [32].

Furthermore, we assume that each node is aware of the identities of its trusted in-neighboring values. Following the ILC-MSR algorithm, at k-th iterative, each node i executes the following three actions:

1) Receive phase: Node i receives the values $\{x_j^k(T), j \in \mathcal{N}_i\}$ from all its in-neighboring nodes at time T, and then sorts the values in a descending order.

2) Filter phase: Consider the existence of values in the list from the trusted links, let us denote $x_{\mathcal{T},M}$ and $x_{\mathcal{T},m}$ as the maximum and minimum values from the trusted links at time T. If there are F or more values that are larger (smaller) than $x_{\mathcal{T},M}$ ($x_{\mathcal{T},m}$), remove the F largest (smallest) values. Otherwise, node i disregards all larger (smaller) values in the list. If i has received no value from trusted links, just disregards precisely the largest (smallest) F values in the list.
3) Update phase: Then by only using in-neighboring information of each node, we give the control input of node i designed as

$$u_i^{k+1} = u_i^k + \gamma_i \sum_{j \in \mathcal{R}_i^k} b_{ij}^k (x_j^k(T) - x_i^k(T)), \tag{3}$$

where $\gamma_i \in \mathbb{R}_{>0}$ is a positive learning gain to be designed, $\mathcal{R}_i^k$ is the set of all remaining nodes that survive the filter phase, and b_{ij}^k is the normalizing weight given to edge e_{ij}, which is given by

$$b_{ij}^k = \begin{cases} \dfrac{a_{ij}}{\sum\limits_j a_{ij}} & j \in \mathcal{R}_i^k; \\ 0 & j \notin \mathcal{R}_i^k. \end{cases} \tag{4}$$

Remark 4: In our algorithm, in order to mitigate the influence of deception attacks (data falsification attacks) on the system, each sensor node ignores the suspicious values by a local filter and fusion rule, that is, removing at most F largest and smallest values from its neighboring nodes, except for the values from the trusted links. Then the remaining values as the control input to determine the node's state value for the next time step, based on which condition (i) in *Definition* 3 can be guaranteed.

We can equivalently obtain that $x_i^k(t) = x_i^k(0) + tu_i^k$ based on the dynamic of (1). Using this fact that the initial reset condition (2), the state of node i at time T can be expressed by

$$\begin{aligned} x_i^{k+1}(T) &= x_i^k(T) + [x_i^{k+1}(T) - x_i^k(T)] \\ &= x_i^k(T) + [x_i^{k+1}(0) - x_i^k(0)] + T(u_i^{k+1} - u_i^k) \\ &= x_i^k(T) + T(u_i^{k+1} - u_i^k). \end{aligned} \tag{5}$$

Combining (3) and (5), we have

$$x_i^{k+1}(T) = x_i^k(T) + T\gamma_i \sum_{j \in \mathcal{R}_i^k} b_{ij}^k [x_j^k(T) - x_i^k(T)]. \tag{6}$$

The update (6) can be written in matrix form as

$$x^{k+1}(T) = (I - T\Gamma\bar{B}(k))x^k(T), \tag{7}$$

where $\Gamma = diag\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$, $\bar{B}(k) = I - B(k)$, and $B(k) = [b_{ij}^k]$.

The next theorem shows the convergence of this algorithm. Let $\mathcal{G}(k)$ denote the graph after removing a certain number of links in filter phase of ILC-MSR algorithm at the k-th iteration.

Theorem 1: Under Assumption 1, if the network topology of the system (1) with protocol (3) is satisfied $(2F+1)$-robust with trusted links, then the finite-time consensus can be achieved when the positive learning gain satisfy

$$T\gamma_i < 1, \; i \in \mathcal{V}. \tag{8}$$

In order to prove Theorem 1, the next technical lemmas are needed, which are adopted from the literature [35]–[37].

Lemma 5: Suppose $\{S_1, S_2, \ldots, S_k\}$ is a finite set of SIA matrices with the property that every finite matrix product $S_{i_j}S_{i_{j-1}}\cdots S_{i_1}$ (repetitions permitted) is SIA. Then, for each infinite sequence $S_{i_1}, S_{i_2}, \ldots$ (repetitions permitted) there exists a column vector $v \in \mathbb{R}^n$ such that

$$\lim_{k\to\infty} S_{i_j}S_{i_{j-1}}\cdots S_{i_1} = 1_n v^T. \tag{9}$$

Lemma 6: Suppose the union of a set of directed graphs $\mathcal{G}(k_1), \mathcal{G}(k_1+1), \ldots, \mathcal{G}(k_2)$ contains a spanning tree, then the matrix product $\prod_{k_1}^{k_2} \Omega(k)$ is SIA, where $k_2 > k_1$, $\Omega(k)$ is a stochastic matrix corresponding to each digraph $\mathcal{G}(k)$.

Proof of Theorem 1: The proof is divided into two steps. Step 1: We first prove that the condition (i) is satisfied, i.e., each node keeps own state value within the interval $[x_{\min}^0, x_{\max}^0]$. According to the definition of $x_{\max}^k$, for each node $i \in \mathcal{V}$, it following from (6) that

$$\begin{aligned} x_i^{k+1}(t) &= x_i^k(t) + t\gamma_i \sum_{j \in \mathcal{R}_i^k} b_{ij}^k [x_j^k(t) - x_i^k(t)] \\ &\le x_i^k(t) + t\gamma_i \sum_{j \in \mathcal{R}_i^k} b_{ij}^k [x_{\max}^k - x_i^k(t)] \\ &= \alpha x_{\max}^k + (1-\alpha)x_i^k(t) \\ &\le x_{\max}^k, \quad t \in [0, T], \end{aligned}$$

where $\alpha = t\gamma_i < 1$. As a result, we have $x_{\max}^{k+1} \le x_{\max}^k$. Similarly, one can use the same argument to get $x_{\min}^{k+1} \ge x_{\min}^k$, which is omitted here for brevity.

Iterating, we obtain for any k,

$$x_{\min}^0 \le x_{\min}^k \le x_{\max}^k \le x_{\max}^0,$$

which guarantees the safety condition (i).

Step 2: In the following, we will show that the network with the ILC-MSR algorithm will achieve finite-time consensus under the F-local deception attack. Notice that, $I - T\Gamma\bar{B}(k) = I - T\Gamma[I - B(k)] = (I - T\Gamma) + T\Gamma B(k)$. Clearly, $I - T\Gamma$ is a diagonal matrix whose diagonal entries are equal to $1 - T\gamma_i$. That is

$$I - T\Gamma = \begin{bmatrix} 1 - T\gamma_1 & & & \\ & 1 - T\gamma_2 & & \\ & & \ddots & \\ & & & 1 - T\gamma_n \end{bmatrix}. \tag{10}$$

We can see from (10) that $I - T\Gamma$ is a non-negative matrix under the condition (8). On the other hand, since Γ has the property that all of its off-diagonal elements are non-negative, we have $\Gamma \ge 0$. Since $T > 0$, $\Gamma \ge 0$, and $B(k) \ge 0$, according to Horn and Johnson [38], we know that the matrix $T\Gamma B(k)$ is a non-negative matrix. Again with the help of Horn and Johnson [38], one can conclude that $I - T\Gamma\bar{B}(k) = (I - T\Gamma) + T\Gamma B(k)$ is a non-negative matrix. According to the definition of b_{ij}^k, it is obvious that the matrix $\bar{B}(k)$ satisfies $\bar{B}(k)1_n = 0$, which, together with $I1_n = 1_n$, further implies that $[I - T\Gamma\bar{B}(k)]1_n = 1_n$. Thus, $[I - T\Gamma\bar{B}(k)]1_n$ is a stochastic matrix associated with the graph $\mathcal{G}(k)$.

Since the initial graph is $(2F + 1)$-robust with $\mathcal{E}_{\mathcal{T}}$, after removing $2F$ or fewer edges from each node in the filter phase of our algorithm, the remaining graph $\mathcal{G}(k)$ is still 1-robust with $\mathcal{E}_{\mathcal{T}}$ by Lemma 2. Then it follows from Lemma 4 that $\mathcal{G}(k)$ contains a spanning tree. Thus, by Corollary 3.5 in [36], the matrix $I - T\Gamma\bar{B}(k)$ has an eigenvalue $\lambda = 1$ with the algebraic multiplicity equal to one. Since all the diagonal elements of $T\Gamma B(k)$ are always equal to zero, i.e. $T\gamma_i b_{ii}^k = 0, \forall i \in \mathcal{V}$, where $b_{ii}^k = 0$ by definition, this implies that the diagonal elements of $I - T\Gamma\bar{B}(k)$ and $I - T\Gamma$ are the same. It is guaranteed that system matrix $I - T\Gamma\bar{B}(k)$ is a stochastic matrix with positive diagonal elements under the condition (2). Again, from Corollary 3.5 in [36], one can determine that matrix $I - T\Gamma\bar{B}(k)$ has the property that $|\lambda| < 1$ for every eigenvalue not equal to one. From Lemma 3.7 in [36], we have that matrix $I - T\Gamma\bar{B}(k)$ is SIA. It is easy to know that the union of the digraphs $\mathcal{G}(1), \mathcal{G}(2), \ldots, \mathcal{G}(k)$ has a spanning tree since each $\mathcal{G}(k)$ has a spanning tree, which by Lemma 6 implies that the matrix product $[I - T\Gamma\bar{B}(k)]\ldots[I - T\Gamma\bar{B}(2)][I - T\Gamma\bar{B}(1)]$ is SIA. Then by Lemma 5, there exists a column vector $v \in \mathbb{R}^n$ such that

$$\lim_{k\to\infty} [I - T\Gamma\bar{B}(k)]\cdots[I - T\Gamma\bar{B}(2)][I - T\Gamma\bar{B}(1)] = 1_n v^T. \tag{11}$$

Substituting (11) into (7) and calculating $x^k(T)$, the system (1) with protocol (3) is equivalent to

$$\begin{aligned}\lim_{k\to\infty} x^k(T) &= \lim_{k\to\infty} [I - T\Gamma\bar{B}(k-1)] \\ &\quad \cdots [I - T\Gamma\bar{B}(2)][I - T\Gamma\bar{B}(1)]x_0(T) \\ &= 1_n v^T x_0(T).\end{aligned} \tag{12}$$

From (12), we can obtain that the consensus condition (ii) is achieved.

Summarizing, we complete the proof. ∎

Theorem 2: Under Assumption 1, if the network topology of the system (1) with protocol (3) can form a spanning tree only with a subset of $\mathcal{E}_{\mathcal{T}}$, then the finite-time consensus can be achieved when the selected learning gain and T satisfy condition (8).

Proof of Theorem 2: Since the graph $\mathcal{G}$ of system (1) can form a spanning tree only with $\mathcal{V}$ and a subset of $\mathcal{E}_{\mathcal{T}}$, by Lemma 3, which implies that $\mathcal{G}$ is an infinity-robust graph. Then by Lemma 2, we know that $\mathcal{G}$ also satisfies $(2F + 1)$-robust. Finally, by the same argument as the proof of Theorem 1, the conclusion follows. ∎

IV. FINITE-TIME CLOCK SYNCHRONIZATION FOR WSNS

In this section, we apply the ILC-MSR algorithm introduced in the previous section to the clock synchronization problem for WSNs in the presence of deception attacks.

A. CLOCK MODEL

We consider a network with n sensor nodes. Each node has a crystal oscillator, which is used to calculate its own hardware local clock. By refereing to [5], [6], the local clock model for each node can be approximated as a linear model, which is given by

$$H_i(t) = \alpha_i t + \beta_i, \tag{13}$$

where H_i is the hardware clock reading, t is the absolute reference time, α_i is the hardware clock drift which determines the clock speed, and β_i is the hardware clock offset.

Here, we emphasize that in practice the absolute reference time t is not available to all the nodes. Hence, it is not possible to compute and manually adjust the parameters α_i and β_i. In order to synchronize all the nodes with respect to a common clock, the concept of logical clock is introduced to replace the hardware clock, which is given by

$$\bar{H}_i(t) = \hat{\alpha}_i H_i(t) + \hat{\beta}_i, \tag{14}$$

where $\bar{H}_i$ is the logical clock reading, $\hat{\alpha}_i$ and $\hat{\beta}_i$ are two adjusting parameters, which are used to correct the values of α_i and β_i respectively. In the above context, the goal of finite-time clock synchronization under the F-local deception attack models is to find $(\hat{\alpha}_i, \hat{\beta}_i)$ for every node which satisfies

$$\lim_{k\to+\infty} \bar{H}_i^k(T) - \bar{H}_j^k(T) = 0, \quad \forall i, j \in \mathcal{V}. \tag{15}$$

where k is the iteration number. The previous expression can be rewritten by substituting (13) into (14) to get

$$\bar{H}_i(t) = \hat{\alpha}_i\alpha_i t + \hat{\alpha}_i\beta_i + \hat{\beta}_i, \tag{16}$$

where $\hat{\alpha}_i\alpha_i$ and $\hat{\alpha}_i\beta_i + \hat{\beta}_i$ are the logical clock skew and offset, respectively. Then, (15) is equivalent to

$$\begin{cases}\lim_{k\to+\infty} \hat{\alpha}_i^k(T)\alpha_i = \alpha_c, \\ \lim_{k\to+\infty} \hat{\alpha}_i^k(T)\beta_i + \hat{\beta}_i^k(T) = \beta_c, \quad i = 1, 2, \ldots, n,\end{cases} \tag{17}$$

where both α_c and β_c are constants, and k is the iteration number. Eq. (17) guarantees that all sensor nodes will have the common logical clock skew and offset within a finite time T.

Consider two neighboring nodes denoted by i and j, respectively. The relative skew α_{ij} is defined as

$$\alpha_{ij} = \frac{\alpha_j}{\alpha_i}, \quad i, j \in \mathcal{V}. \tag{18}$$

The relative skew plays an important role in a distributed synchronization protocol. However, the value of α_{ij} cannot be computed by (18) directly since the true values of α_i and α_j are unavailable. Fortunately, there is another effective method adopted in [10] and most of the other consensus-based clock synchronization algorithms such as [6], [39] to address this issue. And the relative skew α_{ij} can be estimated by any two pairs of hardware clock reading of i and j. That is,

$$\alpha_{ij} = \frac{H_j(t_1) - H_j(t_0)}{H_i(t_1) - H_i(t_0)}, \quad \forall i, j \in \mathcal{V}, \tag{19}$$

where t_1 and t_0 are two different time instant, $t_1 \neq t_0$.

It should be pointed out that in this paper we assume that the process of message exchange is instantaneous, so that the transmission and communication delays can be ignored. This assumption has been widely adopted in existing works for clock synchronization, e.g., [6], [10].

Remark 5: In this paper, we give a strict condition on the network topology. That is, we assume the sensor network is fixed. Our results cannot extent to the dynamic network model. The main reason is that nodes' entering and leaving of the dynamic network will have an impact on the robustness of the network. From the authors' knowledge, this is also a common assumption in almost all of the existing literature on resilient consensus problem, e.g., [6], [13], [20], [32], [40].

B. ALGORITHM OF CLOCK SYNCHRONIZATION

To achieve the goal (17), we present the following distributed clock synchronization scheme. The basic procedure of our scheme is summarized in the table Algorithm 1.

In Algorithm 1, the system iterates k times and the execute time for each iteration is T. $F \in \mathbb{N}$ is the largest number of compromised edges in each node' neighborhood, and $P \in \mathbb{R}$ is the common update period of each node. Observe that the slight differences of hardware clock skew among nodes, the actual update period P_i of each node should be P/α_i, for $i = 1, 2, \ldots, n$. That means in reality the update process of each node is asynchronous.

During each iteration k, the initial conditions for the adjusting parameters of node i are set to $\hat{\alpha}_i(0) = 1$ and $\hat{\beta}_i(t) = 0$, respectively (Line 1).

TASK 1 defines node's state updating round and is activated periodically whenever node i' own update period P_i arrives (Line 4). Each node maintains 4 sets, which denoted by ξ, ψ, $\mathcal{V}_{\max}$, and $\mathcal{V}_{\min}$, initially set them to empty (Line 6). The main idea of update rule is developed from the ILC-MSR algorithm to calculate the logical skew and logical offset of the node at the next time step.

When node i receives a packet $\delta_{j\to i}$ from its neighbor node j, it reads its own current clock $H_i(t)$ and stores $(H_i(t), H_j(t))$ in its memory. If there is already one record from j in the storage, just discards the previous record.

In order to calculate the relative physical skew by (19), node needs to at least two messages from a same neighbor. Therefore, each node possess two areas to store messages. One is referred as M1 (cache area), which is used to store the most recently received message from each neighbor. It also may contain some false data from the compromised links. The other is referred as M2 (record area), which is used to store the remaining data filtered by the ILC-MSR Algorithm.

Algorithm 1 (ILC-MSR Algorithm For Clock Synchronization)

```
Input: $\hat{\alpha}_i$, $\hat{\beta}_i$, $\gamma_i$, $F$, $P$, $T$
Output: $\bar{H}_i(t)$
1:  $\hat{\alpha}_i \leftarrow 1$, $\hat{\beta}_i \leftarrow 0$;
2:  $P_i \leftarrow P/\alpha_i$;
3:  for $t = 0$ to $T$ do
4:      if $H_i(t)/P_i \in \mathbb{N}^+$ then
5:          TASK 1
6:          $\xi \leftarrow \emptyset$, $\psi \leftarrow \emptyset$, $\mathcal{V}_{\max} \leftarrow \emptyset$, $\mathcal{V}_{\min} \leftarrow \emptyset$.
7:          if both $M1_i[j]$ and $M2_i[j]$ are not empty, $\forall j \in \mathcal{N}_i$ then
8:              $\xi \leftarrow \xi \cup j$;
9:          for $j \in \xi$ do
10:             $\hat{\alpha}_{ij} \leftarrow \frac{\hat{\alpha}_j(t_1)\times(H_j(t_1)-H_j(t_0))}{H_i(t_1)-H_i(t_0)}$;
11:             if $\hat{\alpha}_{ij} > \hat{\alpha}_i$ and $j \notin \mathcal{E}_{\mathcal{T}}$ then
12:                 $\mathcal{V}_{\max} \leftarrow \mathcal{V}_{\max} \cup j$;
13:             if $\hat{\alpha}_{ij} < \hat{\alpha}_i$ and $j \notin \mathcal{E}_{\mathcal{T}}$ then
14:                 $\mathcal{V}_{\min} \leftarrow \mathcal{V}_{\min} \cup j$;
15:         if $|\mathcal{V}_{\max}| > F$ then
16:             discard $F$ nodes with the largest values of $\hat{\alpha}_{ij}$ in the set $\mathcal{V}_{\max}$;
17:         else
18:             $\xi \leftarrow \xi \backslash \mathcal{V}_{\max}$;
19:         if $|\mathcal{V}_{\min}| > F$ then
20:             discard $F$ nodes with the smallest values of $\hat{\alpha}_{ij}$ in the set $\mathcal{V}_{\min}$;
21:         else
22:             $\xi \leftarrow \xi \backslash \mathcal{V}_{\min}$;
23:         let $\psi \leftarrow \xi$ represent the set of all remaining nodes;
24:         for $j \in \psi$ do
25:             $M2_i[j] \leftarrow M1_i[j]$;
26:             $M1_i[j] \leftarrow \emptyset$;
27:         $\hat{\alpha}_i(t^+) \leftarrow \hat{\alpha}_i(t) + \gamma_i \sum_{j\in\psi}(\hat{\alpha}_{ij} - \hat{\alpha}_i(t))$;
28:         $\hat{\beta}_i(t^+) \leftarrow \hat{\beta}_i(t) + \gamma_i \sum_{j\in\psi}(\bar{H}_j(t) - (\hat{\alpha}_i(t^+)\bar{H}_j(t) + \hat{\beta}_i(t)))$;
29:         broadcast $[i, H_i(t^+), \hat{\alpha}_i(t^+), \hat{\beta}_i(t^+)]$;
30:         end TASK 1
31:     if node $i$ receives a message from $j$ then
32:         $M1_i(j) \leftarrow [j, H_j(t), \hat{\alpha}_j(t), \hat{\beta}_j(t)]$.
```

Let us use t_1, t_0 to represent the current time and previous time of message in the memory, respectively. For a node i has kept a previous record $[H_i(t_0), H_j(t_0)]$, then, when it receives a new message from j, the relative skew α_{ij} can be obtained directly by (19).

The method of removing the extremum values in the list can ensure the reliability of the remaining $\hat{\alpha}_{ij}$, i.e., each node has a good false data immunity (Lines 9-22).

Since the nodes with extreme values $\hat{\alpha}_{ij}$ in the set ξ have been removed, the message received from the remaining nodes can directly store in the memory. Then, node i empties the data in M1 and is ready to receive the new data in the next time (Lines 24-26).

Then based on the values of all remaining nodes in the set ψ, node i updates its parameters $\hat{\alpha}_i$ and $\hat{\beta}_i$ as $\hat{\alpha}_i(t^+) \leftarrow \hat{\alpha}_i(t)+\gamma_i \sum_{j\in\psi}(\hat{\alpha}_{ij}(t)-\hat{\alpha}_i(t))$ and $\hat{\beta}_i(t^+) \leftarrow \hat{\beta}_i(t)+\gamma_i \sum_{j\in\psi} (\bar{H}_j(t)-(\hat{\alpha}_i(t^+)\bar{H}_j(t) + \hat{\beta}_i(t)))$, respectively, where t^+ indicates the update, and γ_i is the designed learning gain (Lines 27-28).

Eventually, by the iteration of the algorithm, the logical clock of all nodes will reach a common value.

Applying Theorem 1 to the clock synchronization dynamic model (14), which gives rise to the following result.

Corollary 1: The finite-time clock synchronization problem (17) under the F-local deception attack model, can be properly resolved by applying ILC-MSR based Algorithm 1. If the network topology satisfies $(2F+1)$-robust with trusted links, and the selected learning gain γ_i and iteration time T satisfy condition (8).

V. SIMULATION

In this section, we illustrate our distributed clock synchronization algorithm using two numerical simulations. All simulation experiments have been performed with MATLAB.

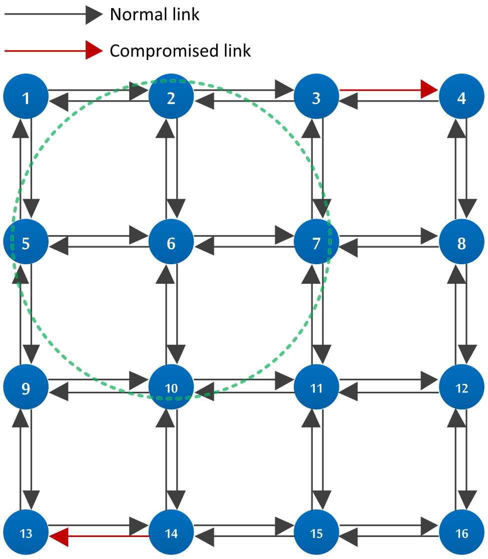

FIGURE 2. WSN with 16 sensor nodes.

A. EXAMPLE 1

We consider 16 nodes connected over a 4 × 4 WSN grid. The distance between adjacent nodes is one unit length. We assume that the communication radius of each node is also one unit length, then the corresponding communication network is shown in Fig. 2, where the green dotted circle represents the communication range of node 6, for example. Then, one can use any existing network robustness determining algorithms [30], [31] (only effective for simple networks) to verify that this network satisfies 1-robust.

Here, we set the common synchronization period of $P = 10s$ and the protocol parameters α_i $(i = 1, \ldots, 16)$ and β_i $(i = 1, \ldots, 16)$ are randomly generated with the MATLAB function 'rand' from the interval [0.6, 1.4] and [0, 100](s), respectively. In particular, we choose the operation time $T = 30s$ for each iteration and let the learning gain γ_i $(i = 1, \ldots, 16)$ be generated with the MATLAB function 'rand' from the interval [0.01, 0.03]. It is clear that the condition (8) can be satisfied by the chosen T and γ_i.

In the simulation, we assume that links (3, 4) and (14, 13) are compromised (red arrow lines) by the deception attacks. And the attacker can randomly modify the logical drift values conveyed in these two links. Let $\alpha'(t)$ denote the modified logical drift value at time t. We select $\alpha'(t) = 1.5$ in this example. Therefore, it is a 1-local deception attack model by the definition.

As the topology of the network is 1-robust, which means that it cannot satisfy the topology condition of Theorem 1, hence the system cannot achieve consensus on this network. The simulation result of the 200th iteration is shown in the Fig. 3. The figure clearly shows that under the interference of false information $\alpha'(t)$, the logical clocks of all nodes cannot reach a synchronization within the fixed time $T = 30s$ (each node's trajectory of logical time is marked with a different color in Fig. 3).

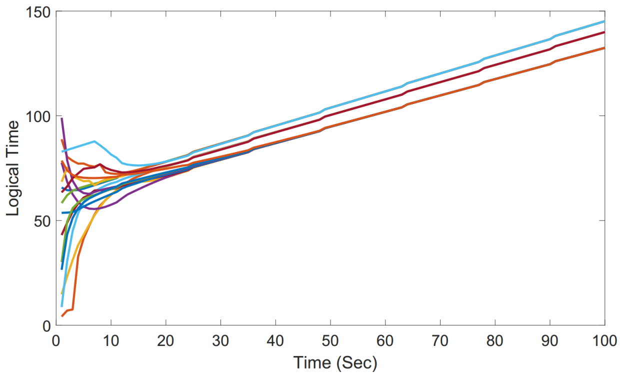

FIGURE 3. The trajectory of each node's logical clock reading of the network under adversaries behavior.

Next, we individually set communication links (1, 5), (2, 1), (3, 2), (4, 3), (5, 9), (8, 4), (9, 13), (12, 8), (12, 11), (13, 14), (14, 15) and (15, 16) as the trusted links (blue arrow lines). The newly generated network with trusted links is shown in Fig. 4.

Then by Definition 6, the network is ensured to be 3-robust with $\mathcal{E}_{\mathcal{T}}$. Therefore, with the help of trusted links, the system satisfies the topology condition of Theorem 1 now. The logical clock trajectory of all nodes at the 200th iteration is show in Fig. 5. We observe that even with the presence of the compromised links, the final clock synchronization is achieved in a fixed time $T = 30s$, which confirms our theoretical predication.

B. EXAMPLE 2

In this example, we consider a WSN with tree structure topology to demonstrate the effect of trusted links on the

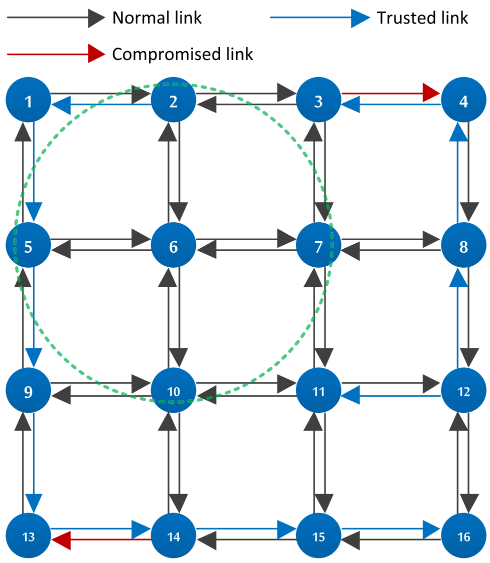

FIGURE 4. Network with trusted links.

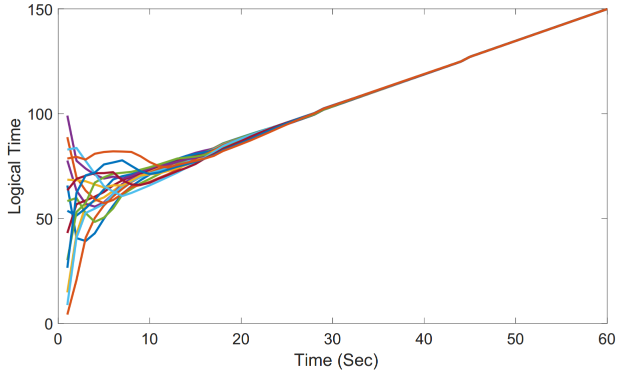

FIGURE 5. The trajectory of each node's logical clock reading of the network with trusted links.

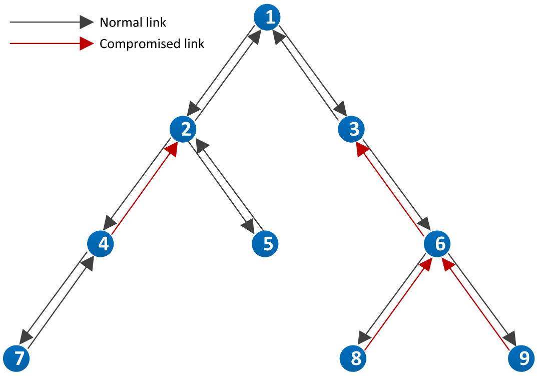

FIGURE 6. WSN communication topology of 9 sensor nodes.

performance of the proposed algorithm. Consider a network consisting of 9 sensor nodes. The communication graph is given as in Fig. 6, where the graph $\mathcal{D}$ contains a spanning tree with a root node labeled 1.

First, we apply the ILC-MSR clock synchronization algorithm with the same parameters α_i $(i = 1, \ldots, 9)$, β_i $(i = 1, \ldots, 9)$ and P as given in Example 1. In this case, we choose

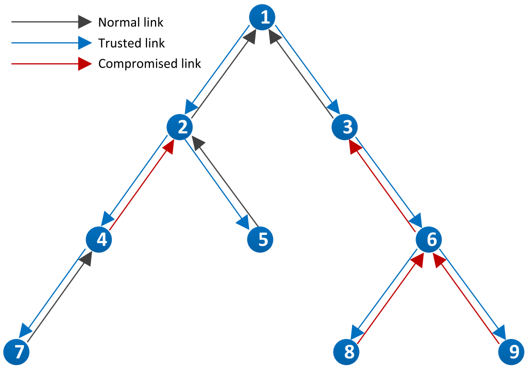

FIGURE 7. WSN contains a spanning tree with trusted links.

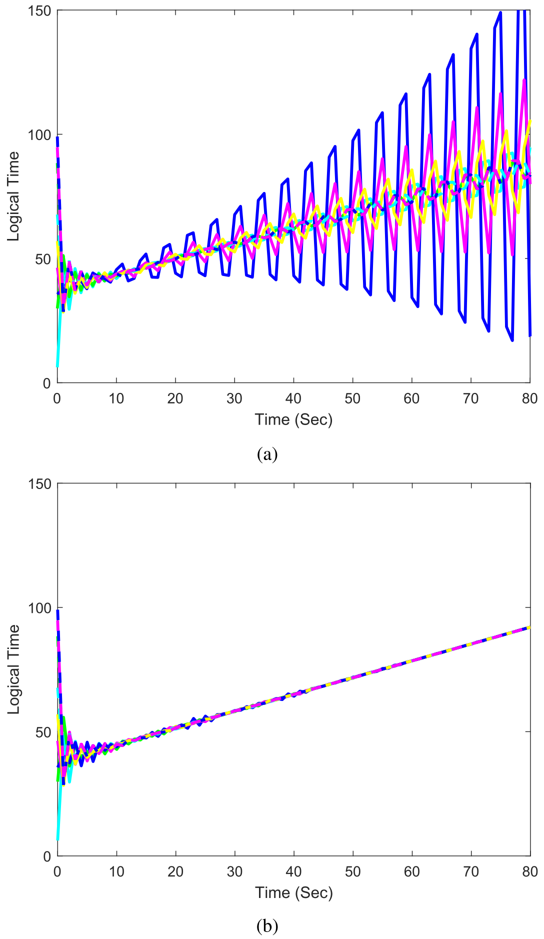

FIGURE 8. (a): Clock synchronization is not achieved for the network in Fig. 6 with nodes under Algorithm 1. (b): Clock synchronization is achieved for the network in Fig. 7 with nodes under Algorithm 1.

the operation time $T = 80s$ and let the learning gain γ_i $(i = 1, \ldots, 9)$ be generated with the MATLAB function 'rand' from the interval $[0, 0.009]$, which also ensures that condition (8) is satisfied. In this case, we select the following false information $\alpha'(t)$:

$$\alpha'(t) = \frac{t}{10}\pi \sin(\frac{t}{10}\pi).$$

Fig. 8(a) presents the simulation result of the 200th iteration. It shows that, as time goes on, the logic clock of the nodes can still not reach a common value in the presence of deception attacks (each node's trajectory of logical time is marked with a different color in Fig. 8). This is because the connectivity of the tree network is not enough for nodes to secure the system when four compromised links are present.

Using Theorem 2, we get that forming a spanning tree only with a subset of $\mathcal{E}_{\mathcal{T}}$ in $\mathcal{D}$ is a sufficient synchronization condition for the network with sparse tree topology. Hence, we choose links $(1, 2), (1, 3), (2, 4), (2, 5), (3, 6), (4, 7), (6, 8)$, $(6, 9)$ as the trusted links, the updated network topology is shown in Fig. 7. Then it can be checked that every sensor node has a directed path from root node 1 only with trusted links. We once again apply the ILC-MSR clock synchronization algorithm with network as show in Fig. 7. The WSN converges to a consensus at a prescribed time $T = 80s$ within 200th iterations as shown in Fig.8(b).

VI. CONCLUSION

This paper presented a new clock synchronization for WSNs under deception attacks, the resilient finite-time clock synchronization, which is based on the ILC-MSR consensus algorithm. We relax the requirement of previous topology condition for resilient consensus by setting a small subset of links trusted, that is, insusceptible to message manipulation attacks. It is proved that under the protocol designed, for a network meets $(2F + 1)$-robust with $\mathcal{E}_{\mathcal{T}}$, the system can mitigate the impact of deception attacks, without isolation of compromised links, and achieve the clock synchronization within the finite time T. The simulation results show the good performance of our approach. In practice, communication delay is a non-negligible constraint in the process of exchanging information for WSNs. Future research efforts will be devoted to the delay tolerant clock synchronization problem of WSNs under attacks, and how to quantify the number of trusted links in a network is another challenging issue to be investigated as future work.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and anonymous reviewers for their very constructive comments, which greatly improved the quality of this article.

REFERENCES

[1] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, "Efficient in-network moving object tracking in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 8, pp. 1044–1056, Aug. 2006.

[2] S. Zhu, C. Chen, J. Xu, X. Guan, L. Xie, and K. H. Johansson, "Mitigating quantization effects on distributed sensor fusion: A least squares approach," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3459–3474, Jul. 2018.

[3] R. Roman, C. Alcaraz, and J. Lopez, "The role of wireless sensor networks in the area of critical information infrastructure protection," *Inf. Secur. Tech. Rep.*, vol. 12, no. 1, pp. 24–31, 2007.

[4] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 281–323, May 2005.

[5] J. He, X. Duan, P. Cheng, L. Shi, and L. Cai, "Accurate clock synchronization in wireless sensor networks with bounded noise," *Automatica*, vol. 81, pp. 350–358, Jul. 2017.

[6] J. He, P. Cheng, L. Shi, J. Chen, and Y. Sun, "Time synchronization in WSNs: A maximum-value-based consensus approach," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 660–675, Mar. 2014.

[7] K. Xie, Q. Cai, and M. Fu, "A fast clock synchronization algorithm for wireless sensor networks," *Automatica*, vol. 92, pp. 133–142, Jun. 2018.

[8] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[9] P. Sommer and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2009, pp. 37–48.

[10] L. Schenato and F. Fiorentin, "Average TimeSynch: A consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, Sep. 2011.

[11] K. Sun, P. Ning, and C. Wang, "TinySeRSync: Secure and resilient time synchronization in wireless sensor networks," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, Alexandria, VA, USA, 2006, pp. 264–277.

[12] H. J. LeBlanc and X. Koutsoukos, "Resilient first-order consensus and weakly stable, higher order synchronization of continuous-time networked multiagent systems," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1219–1231, Sep. 2018.

[13] C. Zhao, J. He, P. Cheng, and J. Chen, "Secure consensus against message manipulation attacks in synchronous networks," in *Proc. World Congr.*, vol. 19, 2014, pp. 1182–1187.

[14] B. Kailkhura, S. Brahma, and P. K. Varshney, "Data falsification attacks on consensus-based detection systems," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 145–158, Mar. 2017.

[15] W. Abbas, A. Laszka, and X. Koutsoukos, "Improving network connectivity and robustness using trusted nodes with application to resilient consensus," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 2036–2048, Dec. 2018.

[16] J. Huang, Y. Wu, L. Chang, M. Tao, and X. He, "Resilient consensus with switching networks and heterogeneous agents," *Neurocomputing*, vol. 341, pp. 70–79, May 2019.

[17] A. Mitra, W. Abbas, and S. Sundaram, "On the impact of trusted nodes in resilient distributed state estimation of LTI systems," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 4547–4552.

[18] Y. Zheng and L. Wang, "Finite-time consensus of heterogeneous multiagent systems with and without velocity measurements," *Syst. Control Lett.*, vol. 61, no. 8, pp. 871–878, Aug. 2012.

[19] M. Franceschelli, A. Giua, and A. Pisano, "Finite-time consensus on the median value with robustness properties," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1652–1667, Apr. 2017.

[20] D. Meng, Y. Jia, and J. Du, "Finite-time consensus for multiagent systems with cooperative and antagonistic interactions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 762–770, Apr. 2016.

[21] Y. Wu, M. Xu, N. Zheng, and X. He, "Attack tolerant finite-time consensus for multi-agent networks," in *Proc. 13th IEEE Int. Conf. Control Automat. (ICCA)*, Jul. 2017, pp. 1010–1014.

[22] J. Usevitch, K. Garg, and D. Panagou, "Finite-time resilient formation control with bounded inputs," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 2567–2574.

[23] D. Meng and Y. Jia, "Finite-time consensus for multi-agent systems via terminal feedback iterative learning," *IET Control Theory Appl.*, vol. 5, no. 18, pp. 2098–2110, Dec. 2011.

[24] H.-S. Ahn and Y. Chen, "Iterative learning control for multi-agent formation," in *Proc. Int. Joint Conf. ICCAS-SICE*, 2009, pp. 3111–3116.

[25] D. Meng, Y. Jia, and J. Du, "Finite-time consensus protocols for networks of dynamic agents by terminal iterative learning," *Int. J. Syst. Sci.*, vol. 45, no. 11, pp. 2435–2446, Nov. 2014.

[26] J. Li and J. Li, "Adaptive iterative learning control for coordination of second-order multi-agent systems," *Int. J. Robust Nonlinear Control*, vol. 24, no. 18, pp. 3282–3299, Dec. 2014.

[27] D. Meng, W. Du, and Y. Jia, "Data-driven consensus control for networked agents: An iterative learning control-motivated approach," *IET Control Theory Appl.*, vol. 9, no. 14, pp. 2084–2096, Sep. 2015.

[28] X. Jin, "Adaptive iterative learning control for high-order nonlinear multiagent systems consensus tracking," *Syst. Control Lett.*, vol. 89, pp. 16–23, Mar. 2016.

[29] H. Zhang and S. Sundaram, "Robustness of information diffusion algorithms to locally bounded adversaries," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 5855–5861.

[30] H. J. LeBlanc and X. D. Koutsoukos, "Algorithms for determining network robustness," in *Proc. 2nd ACM Int. Conf. High Confidence Netw. Syst. (HiCoNS)*, 2013, pp. 57–64.

[31] G. Wang, M. Xu, Y. Wu, N. Zheng, J. Xu, and T. Qiao, "Using machine learning for determining network robustness of multi-agent systems under attacks," in *Proc. 15th Pacific Rim Int. Conf. Artif. Intell.*, Aug. 2018, pp. 491–498.

[32] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.

[33] D. Shen and Y. Wang, "ILC for networked nonlinear systems with unknown control direction through random lossy channel," *Syst. Control Lett.*, vol. 77, pp. 30–39, Mar. 2015.

[34] E. Seneta, *Non-Negative Matrices and Markov Chains*. New York, NY, USA: Springer, 1981.

[35] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proc. Amer. Math. Soc.*, vol. 14, no. 5, pp. 733–737, Oct. 1963.

[36] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[37] F. Xiao and L. Wang, "State consensus for multi-agent systems with switching topologies and time-varying delays," *Int. J. Control*, vol. 79, no. 10, pp. 1277–1284, Oct. 2006.

[38] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[39] Y.-P. Tian, "Time synchronization in WSNs with random bounded communication delays," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5445–5450, Oct. 2017.

[40] D. M. Senejohnny, S. Sundaram, C. De Persis, and P. Tesi, "Resilience against misbehaving nodes in asynchronous networks," *Automatica*, vol. 104, pp. 26–33, Jun. 2019.

YIMING WU received the B.E. degree in automation and the Ph.D. degree in control science and engineering from the Zhejiang University of Technology, Zhejiang, China, in 2010 and 2016, respectively. Between April 2012 and April 2014, he was a Research Assistant with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Since July 2016, he has been with the School of Cyberspace, Hangzhou Dianzi University, Zhejiang. His main research interests include resilient consensus control, iterative learning control, and applications in multiagent systems and sensor networks.

XIONGXIONG HE received the M.S. degree from Qufu Normal University, Qufu, China, in 1994, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 1997. He held a postdoctoral position at the Harbin Institute of Technology, from 1998 to 2000. He joined the Zhejiang University of Technology, Hangzhou, in 2001, where he has been a Professor with the College of Information Engineering. His research interests include nonlinear control, iterative learning control, intelligent control, and applications in multiagent systems and sensor networks.

Dr. He was the General Chair of the 2014 IEEE Conference on Industrial Electronics and the Applications and Technical Program Chair of the 2016 Conference on Data-Driven Control and Learning Systems.