Attacks on Formation Control for Multiagent Systems

Yue Yang, *Student Member, IEEE*, Yang Xiao, *Fellow, IEEE*, and Tieshan Li, *Senior Member, IEEE*

***Abstract*—Multiagent systems (MASs) are distributed systems with two or more intelligent agents. Formation control is a significant control technique of MASs. To date, formation control on MASs is widely used in various fields, such as robots, spacecrafts, satellites, and unmanned aerial/surface/underwater vehicles. However, there is a relatively small body of literature that is concerned with security problems of formation control on MASs in past years. Our research represents the first step toward developing security attacks of formation control on MASs. Our study aims to investigate potential security problems of formation control on a multirobot system for the first time. We propose two kinds of control-level attacks and each kind of attack includes several specific attack forms. Then, we discuss specific features of formation control on a classical multirobot system and utilize theoretical analyses to illustrate how cyberattacks can influence the physical movements of robots. The experimental results of the proposed attacks show that attacks can easily interrupt formation movements of a multirobot system and several carefully designed attacks even can cause irreversible loss.**

***Index Terms*—Cyber attacks, formation control, multiagent systems (MASs), robot mobility.**

I. Introduction

In THE last few decades, there has been a surge of interest in the development of various multiagent systems (MASs) [1], [2]. MASs are distributed systems with two or more intelligent agents [3]. In contrast to a single agent, agents in MASs can cooperatively solve complicated problems with good efficiency, robustness, and reliability.

As one of the most active research topics within the field of MASs, formation control attracts considerable interest since the formation control has widespread applications [2], [4]. Up to now, formation techniques apply to many fields. For spacecrafts or satellites, the formation can allow each agent to stay in a stable distance that can share signal processing and exchange information. For unmanned aerial/surface/underwater vehicles, formation control is widely used in surveillance and searching objects. Aerial vehicle formation can transport goods and overcome the payload limitation of a single aerial vehicle. With the development of formation control, this control technique will be more widely used in various fields in the future. Formation control aims to maintain agents in a predefined shape while the formation moves as a cohesive group.

To maintain a formation shape, agents need to exchange information (e.g., displacement, speed, and/or orientation information) via communication. Wired communication, which is widely used in many control systems, can ensure security by access restrictions since devices in wired networks are physically connected by cables. However, due to the mobility feature of formation control in MASs, wires are unacceptable in many cases. Compared with wired networks, wireless networks have a broadcast nature so that both legitimate and malicious users can access wireless networks [5]. Broadcast communication makes wireless networks more vulnerable than wired networks [6], [7]. A malicious user in a wireless network could modify or disrupt data in communication channels.

Similar to the traditional control systems, MASs are vulnerable to malicious attacks. Thus, far, several preliminary studies about the security issues of MASs appear. LeBlanc and Koutsoukos [8] investigated the robust consensus control for MASs in the situations that a part of agents is compromised. Shames *et al.* [9] focused on detecting, isolating, and removing a fault agent in MASs. With the consideration that compromising communication channels is more common than compromising agents, the literature has grown up around the theme of which one or more communication channels are suffered attacks. Feng *et al.* [10] achieved secure consensus tracking for MASs with connected and disconnected switching topologies caused by connectivity-maintained/broken attacks. Unlike deterministic attacks in [10], Feng *et al.* [11] studied secure consensus tracking for MASs under strategic attacks in cyberspace whose dynamics are captured by a random Markov process. He *et al.* [12] considered secure synchronization of MASs in the situations that sensor-to-controller channels are suffered by false-data injection attacks. Ding *et al.* [13] studied the observer-based event-triggering consensus control problem for MASs with attacks in which can cause probabilistic packet dropouts in a sensor-to-controller channel. Zhu and

Manuscript received April 1, 2021; accepted June 10, 2021. The work of Yue Yang and Tieshan Li was supported in part by the National Natural Science Foundation of China under Grant 51939001 and Grant 61976033; in part by the Science and Technology Innovation Funds of Dalian under Grant 2018J11CY022; in part by the Liaoning Revitalization Talents Program under Grant XLYC1908018 and Grant XLYC1807046; and in part by the Fundamental Research Funds for the Central Universities under Grant 3132019345. This article was recommended by Associate Editor H. Han. *(Corresponding author: Yang Xiao.)*

Yue Yang is with Navigation College, Dalian Maritime University, Dalian 116026, China (e-mail: yueyang@ieee.org).

Yang Xiao is with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487 USA (e-mail: yangxiao@ieee.org).

Tieshan Li is with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China, and also with Navigation College, Dalian Maritime University, Dalian 116026, China (e-mail: tieshanli@126.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3089375.

Digital Object Identifier 10.1109/TCYB.2021.3089375

Martinez [14] proposed a variation of the receding-horizon control protocol to tackle replay attacks in which the control signals are maliciously repeated in controller-to-actuator channels. Receding-horizon control is a method to solve finite time-constrained optimization problems based on iteration. Furthermore, Li *et al.* [15] achieved an event-triggered consensus control for MASs under false-data injection attacks in both sensor-to-controller and controller-to-actuators channels. Zhang *et al.* [16] tackled with resilient control of a networked control system under denial-of-service (DoS) attacks. The closed-loop system in [16] is modeled as a periodic sampled-data control system by introducing a logic processor to capture information about the duration time of each DoS attack. Moreover, Zhang *et al.* [16] utilized a looped functional method to check whether the system can maintain closed-loop stability when under DoS attacks. In [17], an event-triggered load frequency control for MASs under DoS attacks is discussed. Peng *et al.* [17] designed a resilient event-triggering communication scheme to tolerate a degree of packet losses induced by DoS attacks. In [18], a switching-like event-triggered communication scheme is proposed to improve communication efficiency. Compared with the resilient event-triggered communication scheme in [17], the proposed communication scheme can reduce the number of transmitted packets by actively choosing the appropriate event-triggered communication scheme. Shao and Ye [19] and Zhang and Ye [20] dealt with the issue of the secure control design for MASs under DoS attacks. In [19], a fault-tolerant and anti-attack control method is designed by combining a backstepping technique with an event-triggered strategy. Zhang and Ye [20] designed a distributed event-triggered controller with a specific topology to guarantee the system consensus under mode-switching DoS attacks. Based on recent theoretical developments of control issues, a survey of trends and techniques in networked control systems is presented in [21]. Zhang *et al.* [21] discussed recent literature on security control based on data availability and integrity attacks.

As the above literature review, researchers show an increased interest in the security of MASs. However, to date, most literature in the security fields of MAS only focuses on attack detection or attack-resilient control. There is a lack of research about specific system features and attacks based on representative control protocols or applications of MASs. Different systems have different features. In addition, a specific attack also has different effects when the attack launches on different systems or applications. For instance, a DoS attack can keep the windows of a car opening by blocking window systems [22]. Nevertheless, DoS attacks for smart grids can cause a blackout or serious disasters [23], [24] and DoS attacks can also attack domain name systems causing more than half of the Internet of USA to collapse [25]. The above three cases have different hazard levels in realistic. Research on security problems is necessary to consider the specific systems and specific applications. In this article, we aim to investigate system features and attacks based on a classical formation control method of multirobot systems. Although formation control is a significant topic in MAS fields, the security of formation control is still not explored. Before extensively studying attack detection or attack-resilient control methods, we need to investigate potential attacks based on specific features of multirobot formation systems and analyze attack effects theoretically. The contributions of this article are summarized as follows.

1) Previous literature mainly assumes that attackers aim to destabilize systems dynamics by intercepting communication (DoS attacks) or injecting false data (false-data injection attacks). Then, researchers try to overcome attacks through fault-tolerant control or resilient control. In this article, it is the first time in the literature that a series of attacks is designed with a target to formation control protocols on a classical multirobot formation system, and achieve attack purposes without destabilizing systems in a stealthy way. To cause physical effects on robot formation control, the proposed attacks focus on influencing robot displacements and speeds. The displacement attacks can cause robots to deviate from desired positions. Moreover, by elaborately designing, the displacement attacks can hijack robot swarm to a predefined region. For speed attacks, the purposes are increasing energy consumption and reducing work efficiency by slowing robot speeds. We hope that our work can stimulate more research in this direction soon.
2) To show the effects of the proposed attacks, we design the system features and control objectives, and provide theoretical analyses to illustrate how cyberattacks utilize system features to cause an impact on control objectives.
3) We design several experiments to verify the attack effects. The experimental results of the proposed attacks show that attacks can easily interrupt formation movements of a multirobot system and several carefully designed attacks even can cause irreversible loss.

The remainder of this article is organized as follows. In Section II, we describe a classical formation control protocol and a typical MAS. Based on the protocol and the system, we design a series of attacks and analyze these attacks in Sections III and IV, respectively. The attack experimental results are shown in Section V. Finally, conclusions and several future directions are reported in Section VI.

Throughout this article, notations are summarized in Table I.

II. Formation Control on Multiagent Systems

In this article, we consider a robot swarm with N nonholonomic differentially driven mobile robots, where N is the number of robots in formation. The ith robot is modeled by the following nonholonomic dynamic:

$$\begin{cases} \dot{x}^r_{xi}(t) = v^r_i(t)\cos\phi_i(t) \\ \dot{x}^r_{yi}(t) = v^r_i(t)\cos\phi_i(t) \\ \dot{\phi}_i(t) = w_i(t), \\ \dot{v}^r_i(t) = \frac{F_i(t)}{m_i} \\ \dot{w}_i(t) = \frac{\gamma_i(t)}{J_i} \end{cases} \qquad i = 1, 2, \ldots, N \tag{1}$$

where $[x^r_{xi}(t), x^r_{yi}(t)]^T \in \mathbb{R}^2$ is the Cartesian coordinates of the inertial position of the ith robot; $\phi_i(t) \in \mathbb{R}$ is the heading angle; $w_i(t) \in \mathbb{R}$ and $v^r_i \in \mathbb{R}$ are the angular velocity and linear

TABLE I
NOTATIONS

$x^r_{\mathrm{x}i}(t), x^r_{\mathrm{y}i}(t)$	The Cartesian coordinates of the inertial position of the ith robot
$\phi_i(t)$	The heading angle of ith robot
$w_i(t)$	The angular velocity of ith robot
$v^r_i(t)$	The linear velocity of ith robot
$F_i(t)$	The applied force of ith robot
$\gamma_i(t)$	The applied torque of ith robot
m_i	The ith robot mass
J_i	The moment of inertia of ith robot
$x_i(t)$	The hand position/trajectory of robot i
$v_i(t)$, $\dot{x}_i(t)$	The hand speed of robot i
$u_i(t)$, $\ddot{x}_i(t)$	Control input/control protocol of robot i
P_j	Formation patterns
x_{ij}	The desired position of robot i in the formation pattern P_j
$\tilde{x}_{ij}(t)$	The vector from x_{ij} to $x_i(t)$
$E_{gj}(t)$	Goal-seeking error
$E_{fj}(t)$	Synchronization error
$E_j(t)$	The sum of goal-seeking error and synchronization error
K_g, K_f	Parameters of $E_j(t)$ and $u_i(t)$
D_g, D_f	Parameters of $u_i(t)$
$\chi_{ij}(t)$	Displacement-bias attacks
x_a	Displacement bias in displacement-bias attacks
ξ	Path-hijacking attacks
b, α	Parameters of ξ
$\tilde{x}^{in}_{ij}$	Instantaneous position information
η	Destination-hijacking attacks
θ	Desired hijacking position of η
τ	Proportionally reduce-speed attacks
c	Parameters of τ
ε	Gradually reduce-speed attacks
d, e	Parameters of ε
$x^c_a(t)$, $v^c_a(t)$, $u^c_a(t)$	$x_i(t)$, $v_i(t)$, $u_i(t)$ under attacks
$E^c_j(t)$, $E^c_{gj}(t)$, $E^c_{fj}(t)$	Formation errors under attacks

velocity, respectively; $F_i(t) \in \mathbb{R}$ is the applied force; $\gamma_i(t) \in \mathbb{R}$ is the applied torque; $m_i \in \mathbb{R}$ is the ith robot mass; and $J_i \in \mathbb{R}$ is the moment of inertia.

The motion equations (1) can be rewritten as

$$\dot{\vartheta}_i(t) = p(\vartheta_i(t)) + q_i\psi_i(t) \tag{2}$$

where $\vartheta_i(t) = (x^r_{\mathrm{x}i}(t), x^r_{\mathrm{y}i}(t), \phi_i(t), v^r_i(t), w_i(t))^T \in \mathbb{R}^5$, $\psi_i(t) = (F_i(t), \gamma_i(t)) \in \mathbb{R}^2$, $p(\cdot)$ and q_i can be inferred from (1) [26].

As shown in Fig. 1, define a point $x_i(t)$ as a hand position of the ith robot. The point $x_i(t)$ lies a distance l_i along the line that is perpendicular to the wheel axis and intersects $x^r_i(t) = [x^r_{\mathrm{x}i}(t), x^r_{\mathrm{y}i}(t)]^T$. $x_i(t)$ can be presented as follows:

$$x_i(t) = x^r_i(t) + l_i\begin{pmatrix}\cos\phi_i(t)\\ \sin\phi_i(t)\end{pmatrix}. \tag{3}$$

The definition of hand position is useful in some applications. For example, robots are equipped with grippers at their hand positions and robots need to utilize grippers to cooperatively move an object [27].

The system in (2) with output in (3) has constant relative degree and the degree equals to two. The system in (2) can be output feedback linearized about the hand position [26]. Using the diffeomorphism map in [26] and transformed coordinates, the output feedback linearizing control input can be given by

$$\psi_i(t) = \begin{pmatrix}\frac{1}{m_i}\cos\phi_i(t) & -\frac{L_i}{J_i}\sin\phi_i(t)\\ \frac{1}{m_i}\sin\phi_i(t) & \frac{L_i}{J_i}\cos\phi_i(t)\end{pmatrix}^{-1} * \left[u_i(t) - \begin{pmatrix}-v^r_i(t)w_i(t)\sin\phi_i(t) - l_iw_i(t)^2\cos\phi_i(t)\\ v^r_i(t)w_i(t)\cos\phi_i(t) - l_iw_i(t)^2\sin\phi_i(t)\end{pmatrix}\right]$$

where $u_i(t)$ is an additional control input, which can be designed for robot formation. $u_i(t)$ is obtained from (6) and determined by position and speed information of the robot itself and the robot neighbors.

The input–output dynamics of robot hand is described by the following double integrator system [26]:

$$\begin{cases}\dot{x}_i(t) = v_i(t),\\ \dot{v}_i(t) = u_i(t),\end{cases} \quad i = 1, 2, \ldots, N \tag{4}$$

where $x_i(t)$, $v_i(t)$, and $u_i(t)$ denote the position, speed, and control input of robot i, respectively.

Remark 1: In formation control fields, all of the nonholonomic mobile robots, unmanned aerial vehicles, and autonomous underwater vehicles can be described by double integrator systems [28]–[32]. Moreover, Lawton *et al.* [26], Dong *et al.* [28], and Wang *et al.* [29] demonstrated their theoretical results on practical hardware platforms. Therefore, for the rest of this article, we utilize the model (4) to analyze the proposed attacks since we believe that double integrator systems have a wide range of applications. To simple expression, for the rest of this article, we use $x_i(t)$ and $v_i(t)$ to denote positions and speeds of the ith robot, respectively.

We adopt the formation control strategy in [26] and define M formation patterns as $P_j = \{x_{1j}, \ldots, x_{Nj}\}$, where $j = 1, \ldots, M$. For each robot i, there exist $j = 1, \ldots, M$ formation patterns. x_{ij} is the desired position of robot i in the formation pattern P_j. All P_j are prestored in robot memories. Once all robots arrive x_{ij}, the robots update the formation pattern from P_j to P_{j+1} and move toward $x_{i(j+1)}$ in P_{j+1}. Thus, the robots can arrive at a series of the desired positions in order, and the robot trajectories can be planned by designing P_j.

There are two control objectives of robot formation. The first objective is to let robots arrive at desired positions. The second objective is to maintain a fixed formation shape during moving. Define error functions for both two control objectives as follows.

For the first control objective, define $E_{gj}(t) = \sum_{i=1}^{N}\tilde{x}^T_{\mathrm{ij}}(t)K_g\tilde{x}_{\mathrm{ij}}(t)$ as the goal seeking error, where $\tilde{x}_{\mathrm{ij}}(t) = x_i(t) - x_{\mathrm{ij}}$ and K_g is a product of an identity matrix and a positive integer. The subscript character g represents goal seeking. $E_{gj}(t)$ describes the error between current robot positions and the desired robot positions in P_{ij}. Note that $E_{gj}(t) = 0$ if and only if $\tilde{x}_{\mathrm{ij}}(t) = 0$ for all i. Therefore, when $E_{gj}(t) = 0$, all robots arrive at the desired positions in P_{ij}.

For the second control objective, define $E_{fj}(t) = \sum_{i=\langle N\rangle}(\tilde{x}_{\mathrm{ij}}(t) - \tilde{x}_{(i+1)j}(t))^TK_f(\tilde{x}_{\mathrm{ij}}(t) - \tilde{x}_{(i+1)j}(t))$ as the formation synchronization error, where $K_f(t)$ is a product of an identity matrix and a positive integer. The subscript character f represents formation synchronization. The notation

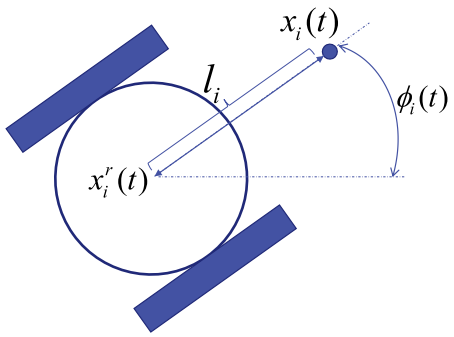


Fig. 1. Nonholonomic differentially driven mobile robot.

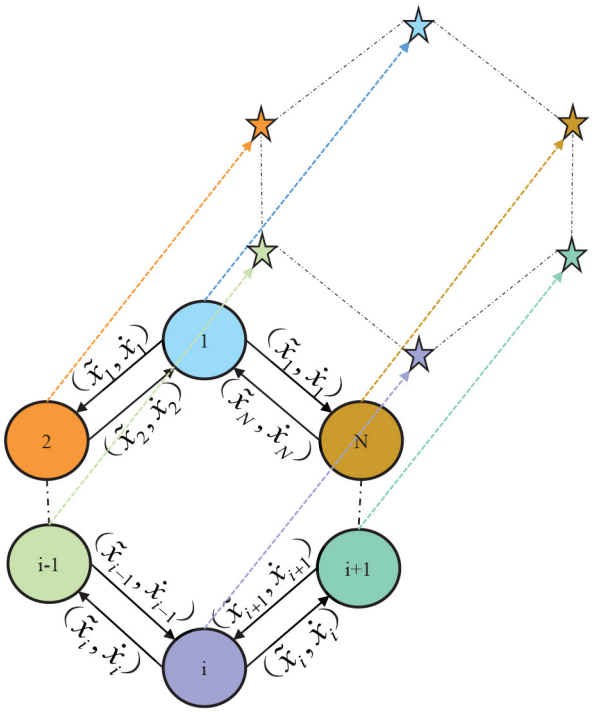


Fig. 2. Formation topology.

$i = \langle N \rangle$ indicates summation around the formation ring, particularly, $\tilde{x}_{(N+1)j} = \tilde{x}_{1j}$. $E_{fj}(t)$ describes the synchronization error between neighboring robots. Note that $E_{fj}(t) = 0$ if and only if $\tilde{x}_{ij} = \tilde{x}_{(i+1)j}$ for all i. Therefore, when $E_{fj}(t) = 0$, all robots maintain the perfect formation shape.

Thus, the total formation error can be presented by $E(t)$

$$\begin{aligned} E_j(t) &= E_{gj}(t) + E_{fj}(t) \\ &= \sum_{i=\langle N \rangle} \Big[\tilde{x}_{ij}^T(t) K_g \tilde{x}_{ij}(t) + \big(\tilde{x}_{ij}(t) - \tilde{x}_{(i+1)j}(t)\big)^T K_f \\ &\quad \times \big(\tilde{x}_{ij}(t) - \tilde{x}_{(i+1)j}(t)\big) \Big] \end{aligned} \tag{5}$$

where K_g and K_f weight the relative importance of goal seeking and formation synchronization. The subscript character g and f represent goal seeking and formation synchronization, respectively. The formation control objective is to derive $E_j(t)$ to zero asymptotically, for all $j = 1, \ldots, M$.

A bidirectional ring topology of the mobile robot swarm is shown as Fig. 2. Stars in Fig. 2 represent the desired positions of the robots. Each robot needs to transmit $\tilde{x}_{ij}(t)$ and $\dot{x}(t)_i$ information to its neighbors. In order to let (5) converge to zero asymptotically, Lawton *et al.* [26] proposed a control protocol

$$\begin{aligned} u_i(t) = &-K_g \tilde{x}_{ij}(t) - D_g \dot{x}_i(t) \\ &- K_f\big(\tilde{x}_{ij}(t) - \tilde{x}_{(i-1)j}(t)\big) - D_f(\dot{x}_i(t) - \dot{x}_{i-1}(t)) \\ &- K_f\big(\tilde{x}_{ij}(t) - \tilde{x}_{(i+1)j}(t)\big) - D_f(\dot{x}_i(t) - \dot{x}_{i+1}(t)) \end{aligned} \tag{6}$$

where K_g, D_g, K_f, and D_f are the products of identity matrices and different positive numbers. In (6), the first two terms are used to let the robot i arrive the desired position in P_{ij}. The third and fourth terms are used to maintain formation with robot $i-1$. The last two terms are used to maintain formation with robot $i+1$.

III. Proposed and Adopted Attacks

Based on the system and the control technique in Section II, we propose two kinds of attacks. One kind of proposed attacks focuses on attacking robot displacement and the other kinds of proposed attacks focus on attacking robot speeds. Each kind of proposed attacks includes several attack forms. Moreover, to achieve the proposed attacks, we adopt man-in-the-middle (MITM) attacks to compromise the communication channels. We further adopt and discuss jamming attacks to directly block communication. The proposed attacks and adopted attacks are summarized and shown in Fig. 3.

To implement proposed attacks, we have several basic assumptions as follows.

Basic Assumptions:

1) *A1:* An attacker has basic abilities of calculation, information storage, and communication;
2) *A2:* The attacker can intercept the communication channel between two robots; the attacker can modify or replace control information and transmit the elaborate control information to one robot;
3) *A3:* Before attacks, robots keep the desired formation shape and move toward predefined destinations.

Remark 2: To cause physical effects on robot formation control, the proposed attacks focus on influencing robot displacements and speeds. The displacement attacks can cause robots to deviate from desired positions. Moreover, by elaborately designing, the displacement attacks can hijack robot swarm to a predefined region. For speed attacks, the purposes are increasing energy consumption and reducing work efficiency by slowing robot speeds.

A. Displacement Attacks

In this section, we design a series of displacement attacks. The displacement attacks can be further classified into displacement-bias attacks and displacement-hijacking attacks.

1) Displacement-Bias Attacks: Displacement-bias attacks aim to achieve that the entire formation cannot arrive the desired positions by injecting displacement biases. Consider that the communication channel from the robot i to the robot $i-1$ is compromised. A basic kind of displacement-bias attacks is that an attacker injects a constant displacement bias x_a into $\tilde{x}_{ij}(t)$, where x_a is a vector and not parallel with vector $\tilde{x}_{ij}(t)$.

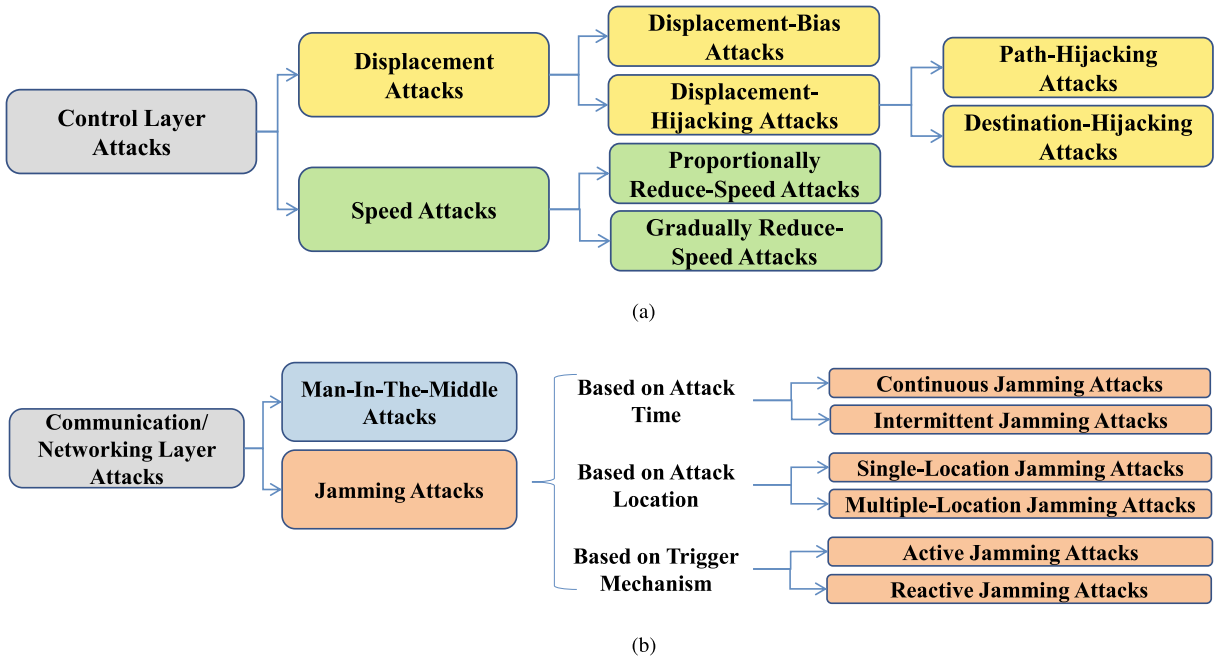


Fig. 3. Classification of formation control attacks. (a) Proposed attacks. (b) Adopted attacks.

Then, the attacker transmits the modified position information $\chi_{ij}(t) = \tilde{x}_{ij}(t) + x_a$ to robot $i-1$.

In the situations without the constant displacement-bias attacks, robot i transmits $\tilde{x}_{ij}(t)$ to robot $i-1$. $\tilde{x}_{ij}(t)$ vectors can decrease to zero when robot i arrives to the desired position. When the attack is launched, the modified vector $\chi_{ij}(t)$ is transmitted to the robot $i-1$ and $\chi_{ij}(t)$ vectors cannot decrease to zero since the x_a vector always exists. Therefore, robot $i-1$ cannot arrive its desired positions since the displacement of robot $i-1$ is effected by $\chi_{ij}(t)$. The displacement of robot $i-1$ has further impacts on the displacement of robot $i-2$ and i. All robots finally have a constant deviation from desired positions.

In addition, we propose two general variant displacement-bias attacks as follows.

1) *Time-Varying Displacement-Bias Attacks:* A time-varying attack modifies displacement information by $\lambda(t)$ with time, where $\lambda(t)$ can be a continuous or a discrete function

$$\chi_{ij}(t) = \tilde{x}_{ij}(t) + \lambda(t)x_a. \tag{7}$$

2) *Random Displacement-Bias Attacks:* A random attack is that injected displacement can randomly be changed within domain (x_a^l, x_a^u), where x_a^l and x_a^u denote the lower and upper magnitude bounds of the bias, respectively

$$\chi_{ij}(t) = \tilde{x}_{ij}(t) + \text{rand}\left(x_a^l, x_a^u\right). \tag{8}$$

Equation $\chi_{ij}(t) = \tilde{x}_{ij}(t) + x_a$ is a particular case of (7) and (8) when $\lambda(t) = 1$ and $x_a^l = x_a^u$, respectively.

The effects of displacement-bias attacks are analyzed in the next section.

2) Displacement-Hijacking Attacks: For Displacement-hijacking attacks, the displacement information $\tilde{x}_{ij}(t)$ are intercepted and replaced by attack messages to hijack the entire robot swarm. Similar to displacement-bias attacks, displacement-hijacking attacks also cause that robots cannot arrive at desired positions. However, attacks messages of displacement-hijacking attacks are carefully designed to achieve hijacking purposes, which have significant and negative effects.

Displacement-hijacking attacks aim to change movements of a robot swarm and remotely control the robot swarm so that attackers can capture the entire robot swarm or let the robot swarm collide with obstacles (e.g., walls or rocks).

Based on attack effects, displacement-hijacking attacks can be further divided into path-hijacking attacks and destination-hijacking attacks. Path-hijacking attacks directly control the movement directions and distance of robots. Attackers modify the path of robots by sequential different attack messages. Destination-hijacking attacks do not concern the path of robots but achieve that robots directly arrive at a predefined region by a constant attack message.

Consider that a robot formation swarm, as described in Section II, is under the path-hijacking attacks. Once the attacker compromises an ongoing communication between robot i and $i-1$, the attacker records the instantaneous position information $\tilde{x}_{ij}^{in}$. Based on the instantaneous position information, the attacker can calculate and derive the attack messages ξ

$$\xi = b\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}\tilde{x}_{ij}^{in} \tag{9}$$

where $b \in (0, \infty)$, $\alpha \in (-\pi, \pi)$, and $\alpha \neq 0$. α is used to change movement directions of the robot swarm and b is used to change movement distance of the robot swarm.

Then, the attacker transmits the attack messages ξ to the robot $i-1$ and does not need to modify any other control messages. After receiving the attack messages, the movement

direction and distance of the robots will change. Moreover, the attacker can adjust b and α and sequentially transmit different ξ. Then, the path of the robot swarm can be controlled by attackers.

For destination-hijacking attacks, if an attacker can obtain the position information $x_i(t)$ of robot i, then the destination-hijacking attacks can be implemented. Unlike path-hijacking attacks, which can change robot paths, the destination-hijacking attacks can directly hijack the robot swarm to a predefined destination region. Suppose that the center of the predefined destination region is at the point θ. Once the attacker compromises an ongoing communication between robot i and $i-1$, the attacker can obtain $\tilde{x}_{\mathrm{ij}}(t)$ from robot i. Based on $x_i(t)$, the attacker can calculate $x_{\mathrm{ij}} = x_i(t) - \tilde{x}_{\mathrm{ij}}(t)$. Then, the attacker transmits the attack message η to robot $i-1$

$$\begin{aligned}\eta &= \theta - \left(x_i(t) - \tilde{x}_{\mathrm{ij}}(t)\right)\\ &= \theta - x_{\mathrm{ij}}.\end{aligned} \tag{10}$$

After receiving the attack message η, robot $i-1$ needs to adjust its positions to keep synchronization with η. Since the physical meaning of η is that robot i stays at position θ, the robot $i-1$ and the entire robot swarm move toward θ.

The purpose of two kinds of hijacking attacks is the same and both aim to remotely control the movement of robots. The difference between two kinds of hijacking attacks is that ξ can directly control the path of robots, whereas η can directly control robots to a predefined destination region but cannot determine the path of how to arrive at the region.

The effects of displacement-hijacking attacks are analyzed in the next section.

B. Speed Attacks

Suitable speeds can ensure that robots fulfill tasks in time. As described in Section II, robot speeds are determined by the control protocol (6). Position information $\tilde{x}_{\mathrm{ij}}(t) = x_i(t) - x_{\mathrm{ij}}$ has further impacts on control protocol (6). For the original system without attacks, when robots move toward their desired position x_{ij}, the robot speeds decrease since magnitudes of robot speeds are determined by $\tilde{x}_{\mathrm{ij}}(t)$ magnitudes. Once all robots arrive x_{ij}, they update formation patterns and generate new speeds to move toward $x_{i(j+1)}$.

The speed attacks aim to reduce work efficiency and increase energy consumption by slowing robot speeds. Once an attacker compromises an ongoing channel between robot i and robot $i-1$, the attacker modifies speed information and transmits attack messages (modified speed information) to robot $i-1$. Based on different attack effects, we design two kinds of attack messages τ and ε

$$\tau_{\mathrm{i}} = c\dot{x}_i(t)$$

where $c \in (0, 1)$

$$\varepsilon_{\mathrm{i}} = ed^t\dot{x}_i(t)$$

where $d \in (0, 1)$ and $e \in (0, \infty)$.

The attack messages τ_i aim to reduce the speed proportionally, and c is a parameter to determine reduced proportion. The attack messages ε_i aim to gradually reduce speeds with time.

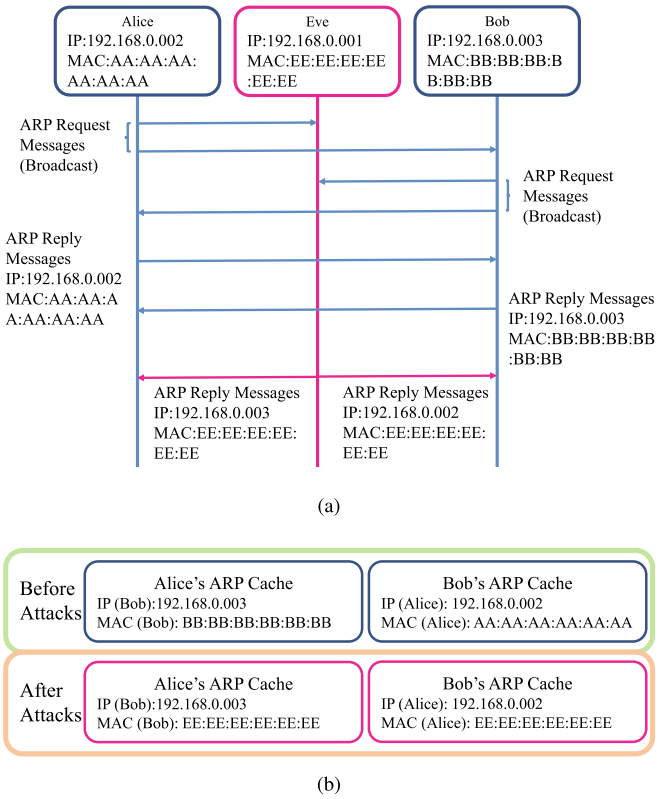


Fig. 4. MITM attacks. (a) Exchanged ARP messages. (b) ARP table.

When $d \in (0, 1)$, the exponential function d^t is asymptotic to zero, where d is used to adjust asymptotic rate and t is a time parameter. e can change the initial value of d^t.

Once robot $i-1$ receives attack messages, it tries to keep pace with the modified speed information. Meanwhile, the other robots also try to maintain synchronization with robot $i-1$. Finally, the entire robot swarm is influenced by attack messages. The speed attacks are stealthy since these attacks do not disrupt tasks directly. Under speed attacks, a robot swarm keeps performing tasks, but robots need to spend more time and energy than usual.

The effects of speed attacks are analyzed in the next section.

C. Compromising Communication Channels

To achieve the above attacks, we adopt an MITM attack to intercept, modify, change, or replace target robots' communication traffic [33].

Mobile robots can communicate with each other through Bluetooth, Wi-Fi, near-field communication (NFC), long-term evolution (LTE), and ZigBee. The MITM attack can launch in all the above communication channels [33]–[35]. Suppose that robots' formation in Section II communicates via Wi-Fi. We explain how the MITM attack compromises communication channels in Wi-Fi networks as an example. MITM attacks work on other networks are similar.

In the Wi-Fi networks, each robot has an Internet protocol (IP) address to indicate source and destination addresses. When robots send data over Wi-Fi, an IP address must be mapped to a media access control (MAC) address since the

MAC address is used in IEEE 802 networking technologies. A MAC address is a unique identifier assigned to a network interface controller. An address resolution protocol (ARP) is used to map an IP address to a MAC address and works on modern Wi-Fi networks. The MITM attacks can be achieved by spoofing ARP. How to spoof ARP and launch MITM attacks are illustrated as follows.

Consider a network: the attacker Eve (IP = 192.168.0.001, MAC = EE:EE:EE:EE:EE:EE), legitimate robot Alice (IP = 192.168.0.002, MAC = AA:AA:AA:AA:AA:AA), legitimate robot Bob (IP = 192.168.0.003, MAC = BB:BB:BB:BB:BB:BB).

First, Alice sends the ARP request message to all the other robots on the network by broadcast, "send the MAC address of 192.168.0.003 to 192.168.0.002." All robots receive the ARP request, but only Bob answers the MAC address in a unicast ARP reply message with the IP-MAC address pair. Alice updates the ARP cache after receiving the ARP reply from Bob. However, ARP is a stateless protocol and stateless protocols handle each request–response pair as an independent event. Alice automatically updates entries in the cache after receiving every ARP replies even though Alice never sends a corresponding ARP request. Thus, cache entries can be easily fabricated by Eve since ARP lacks a proper authentication mechanism [36]. This vulnerability causes an attack, called ARP spoof (also known as ARP cache poisoning).

Based on the ARP spoof technique, Eve sends the ARP replay message (IP = 192.168.0.003, MAC = EE:EE:EE:EE:EE:EE) to Alice and the ARP replay message (IP = 192.168.0.002, MAC = EE:EE:EE:EE:EE:EE) to Bob, as shown in Fig. 4(a). Then, Alice and Bob update their ARP cache, as shown in Fig. 4(b). When Bob or Alice wants to send a message to the other, the message will be sent to Eve. Eve can modify the message before forwarding the message to Bob or Alice. The MITM attack is launched in the communication channel between Alice and Bob.

D. Jamming Attacks

We can also directly block robot communication through jamming attacks. Jamming attacks are a kind of denial of service attacks and can prevent robots from communicating by keeping the communicating medium busy [37], [38]. One drawback of jamming attacks is that they lack stealthiness since the caused problems are easily noticed.

In this section, we adopt jamming attacks to block communication among robots. The robots can utilize real-time control information before being attacked. Once the jamming attacks are launched, robots on both ends of the jammed communication only use the last control information that is stored in robot memories before being attacked.

To understand the impacts of jamming attacks on control protocols, we explain a basic case in which the communication between two robots is continuously jammed. Once the attacker jams the communication between robot i and robot $i-1$ at time n, the robot i and $i-1$ cannot update control information from each other. During the jamming period, the robot i uses $\tilde{x}_{(i-1)j}(n)$ and $\dot{x}_{i-1}(n)$ to calculate its control protocol (6). The robot $i-1$ is similar and uses $\tilde{x}_{(i)j}(n)$ and $\dot{x}_i(n)$ during the jamming period. $\tilde{x}_{(i-1)j}(n)$, $\dot{x}_{i-1}(n)$, $\tilde{x}_{(i)j}(n)$, and $\dot{x}_i(n)$ are constant and cannot update with time. Therefore, the impacts of jamming attacks on the control protocol are equivalent to that an attacker transmits a pair of attack messages $\tilde{x}_{(i-1)j}(n)$ and $\dot{x}_{i-1}(n)$ to robot i, meanwhile, also transmits $\tilde{x}_{(i)j}(n)$ and $\dot{x}_i(n)$ to robot $i-1$.

Extending the above basic case, attackers can launch jamming attacks intermittently since a kind of jamming attack switches between the sleep state and work state to save attackers' energy. The intermittent jamming attacks block communication during work state. For each blocking interval, the robots, which are affected by attacks, use the last control information before blocking communication. When a blocking interval is over, robots try to restart communication with neighbors and update control protocols. If the communication can be restored before the next blocking intervals, robots would keep on moving toward their desired positions and recover their formation shape.

To improve jamming effectiveness, jamming attacks can be launched at different locations of a robot swarm. Multiple-location jamming attacks cause impacts on control protocols and are similar to the basic case. Each robot, in which communication is blocked, uses the last control information before blocking.

Trigger mechanisms of jamming attacks can be divided into active jamming attacks and reactive jamming attacks. The active jamming attacks are launched by attackers on purpose and the reactive jamming attacks are launched when attack targets are detected. The two trigger mechanisms can be applied to different situations. To achieve jamming attacks, we can remotely control one or more malicious robots to get close to the legitimate robots and launch active jamming attacks. Another strategy is that putting malicious robots on the path ahead of the legitimate robots. When the malicious robots observe the network activity of the legitimate robots, the malicious robots launch reactive jamming attacks.

IV. System and Attack Analysis

In this section, we first study the system in Section II to help understand attack mechanisms. Then, we analyze how the displacement and speed attacks influence robot trajectories and speeds.

A. System Analysis

Every kind of control systems has their own specific features. For the formation control system in Section II, four significant features are summarized as follows.

1) *First Feature Is Wireless Networks:* Since formation multirobots cooperatively move to perform tasks, they need to communicate with each other through wireless networks in most situations. Wireless networks have broadcast characters. The broadcast characters cause wireless communications more vulnerable than wired communications to malicious attacks.
2) *Second Feature Is Control Information:* Control information is vital for every control system. Traditional

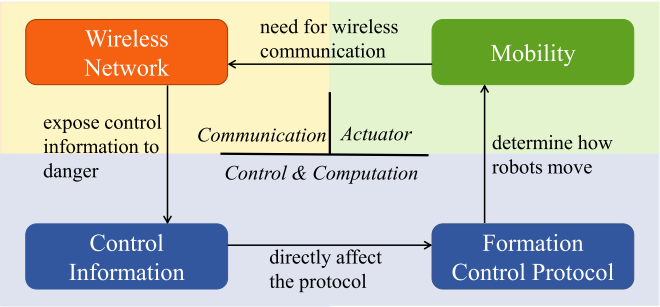


Fig. 5. Features.

industry systems directly transmit control information from controllers to actuators. However, in multirobot formation systems, robots not only transmit control information to their actuators but also need to share a part of control information with other robots through wireless communication.

3) *Third Feature Is the Formation Control Protocol Itself:* The formation control protocol aims to drive robots to the desired positions while maintaining the predefined shape. As shown in (6), the control protocol exchanges $\tilde{x}_{ij}(t)$ and $\dot{x}_i$ to keep a robot synchronize with its neighbors. Once a robot's positions or speeds are affected by attacks, the robot neighbors also need to adjust their positions or speeds to maintain the formation shape.
4) *Fourth Feature Is Robot Mobility:* In multirobot formation systems, mobility is basic and necessary in many applications (e.g., surveillance, searching objects, etc.). Unlike other traditional control systems, once robots are under cyber attacks, robots can be hijacked in the physical sense and be controlled to move to an unknown place. Thus, the robot owners do not have a chance to repair robots and lost the robots forever.

The above four features are closely related, as shown in Fig. 5. Since wireless networks have broadcast nature, control information is exposed to malicious users. If malicious attackers access a communication channel, they can modify or replace control information and transmit unreliable control information to one robot. Then, the robot trajectories or speeds change. Due to formation control protocols, the robot further causes impacts on its neighbor robots. Mobility is a necessary character for many formation applications and results in the need for wireless communication.

In addition to the above features, control objectives are also worth mentioning. According to (5), there are two control objectives. The first objective is that let all robots arrive desired positions. The second objective is to keep robots synchronization. The objectives are achieved based on the control protocol (6). Therefore, the control protocol has two abilities: 1) goal seeking and 2) formation maintenance.

B. Attack Analyses

In this section, we analyze that cyber attacks on the control protocol can cause what impacts on control objectives by utilizing the above features.

When the magnitude or direction of information $x_i(t)$, $v_i(t)$, or $u_i(t)$ is influenced by attack messages, we add a superscript c on the information, such as $x_i^c(t)$. Consider that an attacker transmits attack messages to robot a. The attack messages directly poison the control protocol of robot a. Then, the control protocol of robot a contains the attack messages and can be rewritten as $u_a^c(t)$. Based on the (4), the attack messages further influence robot speeds and positions. We can obtain $x_a^c(t)$ and $\dot{x}_a^c(t)$ in which also include a part of attack messages. Then, robot a transmits $x_a^c(t)$ and $\dot{x}_a^c(t)$ to robot $a+1$ and robot $a-1$. Therefore, the control protocol of robot $a+1$ and robot $a-1$ is also poisoned by attack messages. By the analogy, the control protocols of all robots are influenced by attack messages and the entire double integrator system can be rewritten as

$$\begin{cases} \dot{x}_i^c(t) = v_i^c(t), \\ \dot{v}_i^c(t) = u_i^c(t), \end{cases} \quad i = 1, 2, \ldots, N. \tag{11}$$

Based on (11), we can rewrite the control protocol (6) and the formation error equation (5) as

$$\begin{aligned} u_i^c(t) = &-K_g \tilde{x}_{ij}^c(t) - D_g \dot{x}_i^c(t) \\ &- K_f\left(\tilde{x}_{ij}^c(t) - \tilde{x}_{(i-1)j}^c(t)\right) - D_f\left(\dot{x}_i^c(t) - \dot{x}_{i-1}^c(t)\right) \\ &- K_f\left(\tilde{x}_{ij}^c(t) - \tilde{x}_{(i+1)j}^c(t)\right) - D_f\left(\dot{x}_i^c(t) - \dot{x}_{i+1}^c(t)\right) \end{aligned} \tag{12}$$

where $\tilde{x}_{ij}^c(t) = x_i^c(t) - x_{ij}$

$$\begin{aligned} E_j^c(t) &= E_{gj}^c(t) + E_{fj}^c(t) \\ &= \sum_{i=\langle N\rangle}\left[\tilde{x}_{ij}^c(t)^T K_g \tilde{x}_{ij}^c(t) + \left(\tilde{x}_{ij}^c(t) - \tilde{x}_{(i+1)j}^c(t)\right)^T K_f \right. \\ &\qquad \left. \times \left(\tilde{x}_{ij}^c(t) - \tilde{x}_{(i+1)j}^c(t)\right)\right]. \end{aligned} \tag{13}$$

For the original system, the control protocol (6) aims to drive robots to move to a series of desired positions and maintain a formation shape. The control objectives are presented by error equation (5). When the error equation (5) is derived to zero asymptotically by the control protocol (6), the robots can satisfy control objectives.

Similar to the control protocol (6) and error equation (5), the new error equation (13) also can be derived to zero asymptotically by the new control protocol (12). However, the meaning of new error equation (13) is different with original error equation (5). Based on the new error equation (13), we analyze whether control objectives still can be achieved under attacks. We provide two theorems for displacement attacks and speed attacks, respectively.

Theorem 1: Based on basic Assumptions A1–A3, under displacement attacks, the trajectories of robots are deviated H_i from the original trajectories, where H_i are deviated displacement vectors for each robot i. H_i are nonzero and not parallel with vectors $\tilde{x}_{ij}(t)$. The robots cannot arrive predefined x_{ij} in formation patterns P_j, but can still keep the formation shape.

Proof of Theorem 1: According to the proposed displacement attacks in Section III, the attack messages are not parallel with original displacement vectors. Suppose that the robot a

directly receives attack messages A_d from robot $a-1$. Based on the control protocol (6), the original control protocol of robot a is presented as follows:

$$\begin{aligned} u_a(t) = &-K_g\tilde{x}_{aj}(t) - D_g\dot{x}_a(t) \\ &- K_f\big(\tilde{x}_{aj}(t) - \tilde{x}_{(a-1)j}(t)\big) - D_f(\dot{x}_a(t) - \dot{x}_{a-1}(t)) \\ &- K_f\big(\tilde{x}_{aj}(t) - \tilde{x}_{(a+1)j}(t)\big) - D_f(\dot{x}_a(t) - \dot{x}_{a+1}(t)). \end{aligned} \tag{14}$$

When displacement vectors $\tilde{x}_{(a-1)j}(t)$ from robot $a-1$ are replaced by attack messages A_d, the original control protocol $u_a(t)$ is poisoned and can be rewritten as follows:

$$\begin{aligned} u_a^c(t) = &-K_g\tilde{x}_{aj}(t) - D_g\dot{x}_a(t) \\ &- K_f\big(\tilde{x}_{aj}(t) - A_d\big) - D_f(\dot{x}_a(t) - \dot{x}_{a-1}(t)) \\ &- K_f\big(\tilde{x}_{aj}(t) - \tilde{x}_{(a+1)j}(t)\big) - D_f(\dot{x}_a(t) - \dot{x}_{a+1}(t)). \end{aligned}$$

Then, we can obtain that

$$\begin{aligned} u_a^c(t) - u_a(t) &= K_fA_d - K_f\tilde{x}_{(a-1)j}(t) \\ &= K_f\big(A_d - \tilde{x}_{(a-1)j}(t)\big). \end{aligned} \tag{15}$$

Let $\ddot{H}_a = K_f(A_d - \tilde{x}_{(a-1)j}(t))$. K_f is a product of an identity matrix and a positive integer. The identity matrix does not change the vector direction of $A_d - \tilde{x}_{(a-1)j}(t)$. Attack messages A_d are not parallel with original displacement vectors $\tilde{x}_{(a-1)j}(t)$. Therefore, $\ddot{H}_a$ is a nonzero vector and not parallel with vector $\tilde{x}_{(a-1)j}(t)$.

As described in basic Assumptions A1–A3, robots keep the desired formation shape before attacks. The displacement vectors and speed vectors in (14) have the same direction and all these vectors are parallel with the original control protocol $u_a(t)$. Therefore, since $\ddot{H}_a$ is not parallel with vector $\tilde{x}_{(a-1)j}(t)$, $\ddot{H}_a$ is not parallel with $u_a(t)$. Then, we can obtain (15) as $u_a^c(t) = u_a(t) + \ddot{H}_a$, where $\ddot{H}_a$ is a nonzero vector and not parallel with vector $u_a(t)$.

The input–output dynamics of each robot is described in (11). Based on the new control protocol $u_a^c(t) = u_a(t) + \ddot{H}_a$, the input–output dynamics of robot a can be rewritten as $\dot{v}_a^c(t) = u_a^c(t) = u_a(t) + \ddot{H}_a$ and $\dot{x}_a^c(t) = v_a^c(t) = v_a(t) + \dot{H}_a$. The new trajectory of robot a is $x_a^c(t) = x_a(t) + H_a$. $\dot{H}_a$ and H_a also include attack messages A_d.

When the robot a shares its new trajectory $x_a^c(t)$ and speed $\dot{x}_a^c(t)$ to its neighbor robots $a+1$ and $a-1$, the control protocols of robots $a+1$ and $a-1$ are also influenced by attack messages A_d. Then, the control protocols of robot $a+1$ and $a-1$ can be rewritten as $u_{a+1}^c(t)$ and $u_{a-1}^c(t)$, respectively. The new control protocols $u_{a+1}^c(t)$ and $u_{a-1}^c(t)$ further influence speeds and trajectories of robot $a+1$ and $a-1$. Then, speeds and trajectories of robot $a+1$ and $a-1$ also include attack messages A_d and are further transmitted to their neighbors. Thus, the attack messages A_d spread among robots.

Under displacement attacks, all robots i have new control protocol $u_i^c(t) = u_i(t) + \ddot{H}_i$, where $\ddot{H}_i$ include attack messages A_d and not parallel with $u_i(t)$. Based on the new control protocol $u_i^c(t)$, all robots have new trajectories $x_i^c(t) = x_i(t) + H_i$, where $x_i(t)$ are original trajectories and H_i are the deviation from original trajectories. Since $\ddot{H}_i$ are nonzero vectors and not parallel with vectors $u_i(t)$, H_i are not parallel with original trajectories $x_i(t)$. The new trajectories $x_i^c(t)$ cannot go through desired positions x_{ij}.

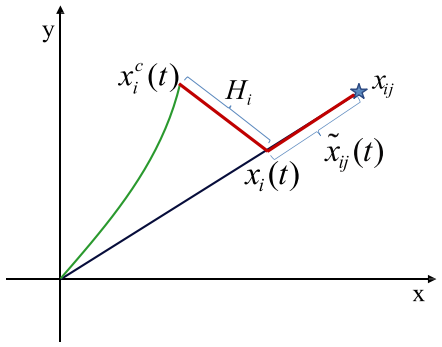


Fig. 6. Proof of Theorem 1.

When the new error equation (13) is derived to zero asymptotically by the new control protocol (12), new goal-seeking error $E_{gj}^c(t)$ and new formation-synchronization error $E_{fj}^c(t)$ are all equal to zero. $E_{fj}^c(t) = 0$ if only if $\tilde{x}_{ij}^c(t) = \tilde{x}_{(i+1)j}^c(t)$ for all i. $\tilde{x}_{ij}^c(t) = \tilde{x}_{(i+1)j}^c(t)$ is equivalent to $x_{ij}^c(t) - x_{(i+1)j}^c(t) = x_{ij} - x_{(i+1)j}$, which is only true when all robots are in the same relative positions. Therefore, the robots still try to keep the formation shape. For $E_{gj}^c(t) = 0$, since the trajectories $x_{ij}^c(t)$ cannot go through x_{ij}, $E_{gj}^c(t) = 0$ will only be true when the magnitude of H_i is equal to the magnitude of $\tilde{x}_{ij}(t)$, as shown in Fig. 6. When $E_{gj}^c(t) = 0$, all robots arrive at positions which are related to H_i. Theorem 1 is proved.

Theorem 2: Based on basic Assumptions A1–A3, under speed attacks, the trajectories of robots are the same as the original trajectories. All robots still can arrive predefined x_{ij} in formation patterns P_j but spend different time compared to the case without attacks. The robots can maintain the formation shape.

Proof of Theorem 2: The speed attacks transmit smaller speed vectors than original speed vectors to one robot. Suppose that robot a directly receives the speed attack messages A_s from robot $a-1$. Similar to the Proof of Theorem 1, the control protocol of robot a also can be rewritten as $u_a^c(t) = u_a(t) + \ddot{H}$ form, where $\ddot{H}_a = D_f(A_s - \dot{x}_{a-1}(t))$. However, since the attack message A_s is parallel with both original speed information $\dot{x}_{a-1}(t)$ and original control protocol $u_a(t)$, $\ddot{H}$ is parallel with the original control protocol $u_a(t)$. Then, the new control protocol can be written as $u_a^c(t) = Q_a u_a(t)$, where Q_a is a scalar. $Q_a \in (0, 1)$ since speed vectors in attack messages are smaller than original speed vectors.

The input–output dynamics of each robot is described in (11). For speed attacks, the input–output dynamics of robot a can be rewritten as $\dot{v}_a^c(t) = u_a^c(t) = Q_a u_a(t)$ and $\dot{x}_a^c(t) = v_a^c(t) = Q_a v_a(t)$. The trajectories are $x_i^c(t) = Q_i x_i(t)$.

When the robots move toward x_{ij} from $x_{i(j-1)}$, the robot trajectories $x_i(t)$ are monotone functions. Since Q_i is a nonzero scalar, there exist $x_i(t+T) = x_i^c(t) = Q_i x_i(t)$, $T > 0$ when $Q_i \in (0, 1)$. $\because x_i^c(t) = x_i(t+T)$, $\therefore \tilde{x}_{ij}^c(t) = \tilde{x}_{ij}(t+T)$. Therefore, the new error equation (13) $E_j^c(t)$ is equal to original error equation (5) $E_j(t+T)$. Since $E_j^c(t)$ can be derived to zero by $u_i^c(t) = Q_i u_i(t)$, the original error equation $E_j(t+T)$ also can be derived to zero by $u_i^c(t) = Q_i u_i(t)$. $E_j(t+T) \to 0$

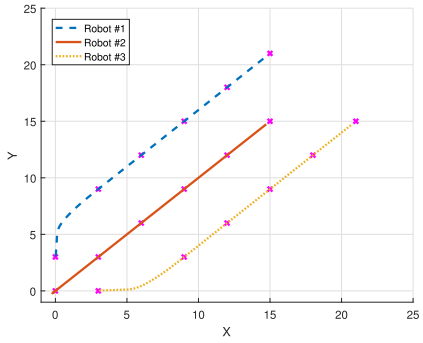


Fig. 7. Trajectories of robots (without attacks).

means that under speed attacks, all robots still can satisfy the two original control objectives but need more time T to arrive desired positions. Theorem 2 is proved.

Remark 3: The above theoretical analyses and theorems are not limited to the attacks in Section III. In the system, the control protocol (6) is a linear equation. When the one of control information $\tilde{x}_{(i-1)j}(t)$, $\dot{x}_{i-1}$, $\tilde{x}_{(i+1)j}(t)$, or $\dot{x}_{i+1}$ is modified or replaced. The control protocol under attacks always can be rewritten as $u_i^c(t) = u_i(t) + \ddot{H}_i$ forms or $u_i^c(t) = Q_i u_i(t)$ forms. Therefore, if researchers propose some new attacks under basic Assumptions A1–A3, they can utilize above theorems to analyze the attacks.

C. Extension of Attack Analyses

To extend attack analyses with a general undirected communication topology, we revise the control protocol (6) as follows.

$$u_i(t) = -K_g\tilde{x}_{ij}(t) - D_g\dot{x}_i(t) - K_f\sum_{k=1}^{N} a_{ik}\big(\tilde{x}_{ij}(t) - \tilde{x}_{kj}(t)\big) - D_f\sum_{k=1}^{N} a_{ik}(\dot{x}_i(t) - \dot{x}_k(t)) \tag{16}$$

where $a_{ik} = 1$ denotes that robot i and robot k exchange control information with each other; otherwise, we have $a_{ik} = 0$. Moreover, $a_{ik} = 1$ indicates that there exists an undirected path between robot i and robot k. To extend attack analyses, we make another assumption as follows.

A4: The undirected communication topology of robot formation is connected, that is, there is an undirected path between each pair of robots.

Remark 4: Assumption A4 is rational since most formation control papers with undirected communication topologies are also based on the same assumption [39]–[41]. Assumption A4 guarantees that each robot can receive information from at least one another robot. To achieve objects of formation control, this assumption is a necessary condition.

When control information $x_i(t)$, $v_i(t)$, or $u_i(t)$ are influenced by attack messages, we add a superscript c on the information, such as $x_i^c(t)$. Consider that a communication channel of robot p is compromised by an attacker and the attacker transmits attack messages to robot p. The attack messages poison the control protocol of robot p and the control protocol can be rewritten as $u_p^c(t)$. Based on (4), we can obtain $x_p^c(t)$ and $v_p^c(t)$ of robot p. Then, the robot p transmits $x_p^c(t)$ and $\dot{x}_p^c(t)$ to its all neighbor robots with $a_{pk} = 1$. Therefore, control protocols of neighbors of robot p are also poisoned by attack messages. Since the communication topology is connected, the influence of attack messages can spread in the robot swarm. The control protocols of all robots are influenced by attack messages and the entire systems can be rewritten as (11).

Corollary 1: If the undirected communication topology of robot formation is connected, Theorems 1 and 2 are still hold.

Proof of Corollary 1: Suppose that a compromised communication channel is between robot p and robot q. An attacker utilizes the compromised communication to transmit displacement attack messages A_d or speed attack messages A_s to robots.

For displacement attacks, the control protocol of robot p can be rewritten as

$$u_p^c(t) = -K_g\tilde{x}_p(t) - D_g\dot{x}_p(t) - K_f a_{pq}\big(\tilde{x}_{pj}(t) - A_d\big) - K_f\sum_{k=1,k\neq q}^{N} a_{pk}\big(\tilde{x}_{pj}(t) - \tilde{x}_{kj}(t)\big) - D_f\sum_{k=1}^{N} a_{pk}\big(\dot{x}_p(t) - \dot{x}_k(t)\big) \tag{17}$$

where $a_{pq} = 1$.

Then, we can obtain that $u_p^c(t) - u_p(t) = K_f(A_d - \tilde{x}_{qj}(t))$, which is the same as (15). The following proof process is the same as proof of Theorem 1. Similarly, for speed attacks, we can obtain $u_p^c(t) - u_p(t) = D_f(A_s - \dot{x}_q(t))$. The following proof process is the same as proof of Theorem 1. Corollary 1 is proved.

V. Numerical Experiment

In this section, the experimental results are provided to demonstrate the original system in Section II and the attack effects in Section III. The experimental results are consistent with the theoretical analyses in Section IV-B.

The parameters of the control protocol (6) are defined as $K_g = 0.5I_2$, $D_g = I_2$, $K_f = 5I_2$, and $D_f = I_2$ [26]. Suppose a scenario that three robots are at initial positions P_0={(0, 3), (0, 0), (3, 0)}. The robots are commanded to move through a series of formation patterns: $P_j = \{(3j-3, 3j+3), (3j-3, 3j-3), (3j+3, 3j-3)\}$, where $j = 1, 2, 3, 4, 5$, and j denotes numbers of formation patterns. The formation patterns update when $E_j(t) < \Omega$. Ω is a threshold value. For all the results shown in this article, $\Omega = 1$.

A. Formation Without Attacks

In Fig. 7, x marks are desired positions of formation patterns. When three robots move without attacks, robots can arrive at a series of formation patterns P_j by using control protocol (6).

B. Displacement Injection Attacks

Three robots intend to move as a trajectory in Fig. 7. However, when the robots arrive at P_2, the communication

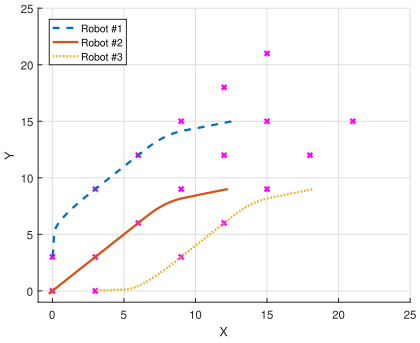


Fig. 8. Displacement injection attacks.

channel between robot #3 and robot #1 is compromised. The displacement vectors from robot #3 is continually injected a vector $x_a = (1, 0)$. Then, the attack messages directly transmit to the robot #1. Since $x_a = (1, 0)$ is not parallel with original displacement vectors, the trajectories of the entire robot swarm are deviated toward the direction that is related to the injected vector, as shown in Fig. 8. The injected vector causes that $E(t)$ cannot decrease small enough to satisfy the threshold value Ω.

C. Path-Hijacking Attacks

Three robots intend to move as a trajectory in Fig. 7. However, when the robots arrive at P_2, the robot #3 continually receives attack messages ξ. To verify the effects of parameters b and α, we design three comparative experiment cases. In case 1, $b = 2$ and $\alpha = \pi/2$, as shown in Fig. 9(a). In case 1, robots deviate some distance to the left direction. In case 2, $b = 2$ and $\alpha = -\pi/2$, as shown in Fig. 9(b). In case 2, robots deviate the same distance as case 1, but toward the right direction. In case 3, $b = 4$ and $\alpha = \pi/2$, as shown in Fig. 9(c). In case 3, robots shift double distance to the left than the distance in case 1. The three comparative cases illustrate that b can change the deviation distance of robots and α can change deviation directions of robots. Therefore, attackers can modify the path of robots through sequential different attack messages ξ.

D. Destination-Hijacking Attacks

Three robots intend to move as a trajectory in Fig. 7. However, when the robots arrive at P_2, the robot #3 continually receives attack messages η. Let the predefined region is a circle with 7-m radius at the center point $\theta = (35, 10)$. Under hijacking attacks with attack messages η, three robots move toward θ and finally, stay around the predefined region, as shown in Fig. 10.

E. Speed Attacks

Before speed attacks, three robot speeds are shown in Fig. 11(a). For the situation without attacks, robot speeds decrease when robots move toward their desired positions x_{ij} since the magnitude of robot speeds is determined by the displacement vectors $\tilde{x}_{ij}(t)$. Once all robots arrive at x_{ij}, they update formation patterns and generate new speeds to move toward $x_{i(j+1)}$. Since the distance between P_1 and P_5 is equal, robots have the same peak values of speeds and spend the same time for each distance interval.

However, when the robots arrive at P_1, the communication channel between robot #1 and robot #3 is compromised.

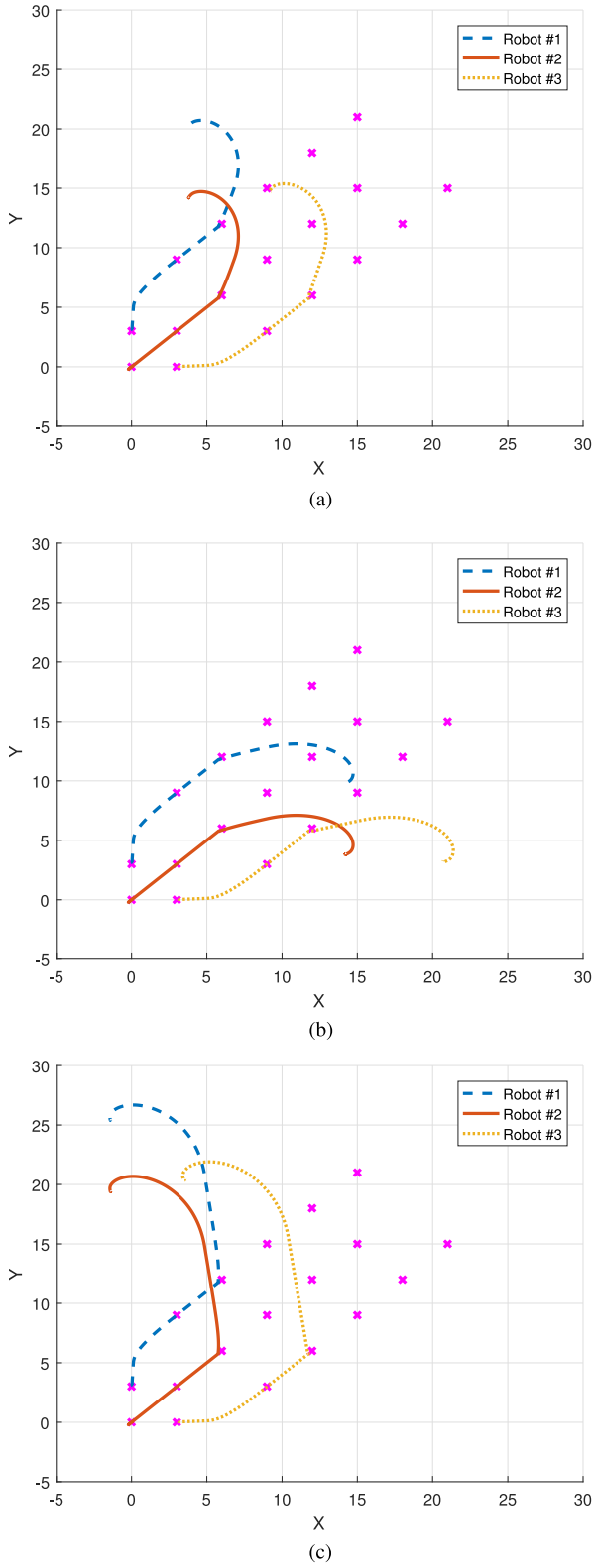


Fig. 9. Path-hijacking attacks. (a) Case 1: $b = 2$ and $\alpha = \pi/2$. (b) Case 2: $b = 2$ and $\alpha = -\pi/2$. (c) Case 3: $b = 4$ and $\alpha = \pi/2$.

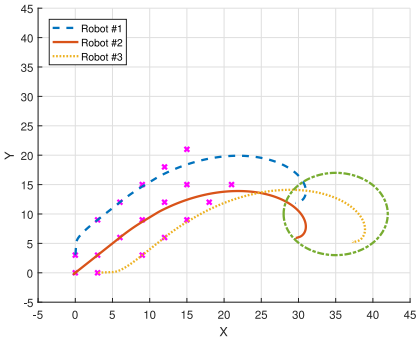
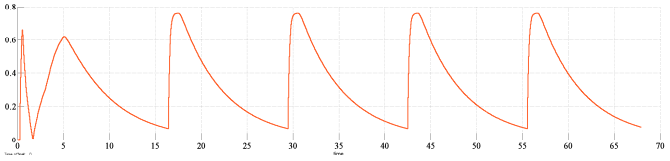
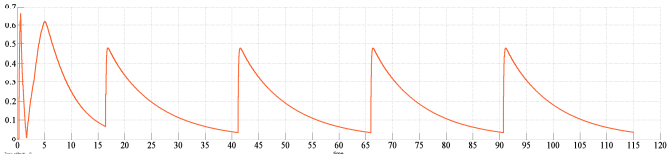


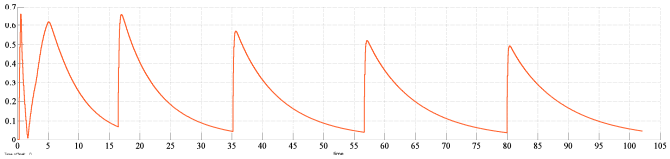
Fig. 10. Destination-hijacking attacks.



(a)



(b)



(c)

Fig. 11. Speeds of robots. (a) Robot speeds without speed attacks. (b) Proportionally reduce-speed attacks. (c) Gradually reduce-speed attacks.

For proportionally reduce-speed attacks, the attacker continually transmits attack messages $\tau = c\dot{x}_1(t)$ to robot #3, where $c = 0.5$. As shown in Fig. 11(b), when robots move from P_1 to P_5 under-speed attacks τ, robots have the same peak values of speeds, which are lower than original systems. The time to reach each formation pattern is the same for each distance interval, but more than original systems.

For gradually reduce-speed attacks, the attacker continually transmits attack messages $\varepsilon = ed^t\dot{x}_1(t)$ to robot #3, where $d = 0.9$ and $e = 0.2$. As shown in Fig. 11(c), when robots move from P_1 to P_5 under speed attacks ε, the peak values of robot speeds are lower and lower. For each distance interval, the robot speeds gradually decrease and the time to reach each formation pattern gradually increases.

VI. Conclusion and Future Work

In this article, we proposed displacement attacks and speed attacks with a target to a classical formation multirobot system. The displacement attacks can remotely change robot trajectories. The speed attacks can reduce robot speeds and increase work time. The proposed attacks further include different attack forms to achieve different attack effects. The displacement-bias attacks can make robots deviate from original trajectories. The displacement-hijacking attacks can hijack robots to achieve remote control. The speed attacks can reduce speeds proportionally or gradually. The proposed attacks are based on MITM attacks. We also adopt and discuss jamming attacks on the system. In addition, to understand how cyber attacks influence robot mobilities, we study the control objectives and system features. Based on theoretical analyses about what are the attack effects on control objectives, we further provide some theoretical studies. The theorems can verify proposed attack effects in theory. We design a series of experiments and the experimental results also match the theorems.

Since most previous researchers suppose that attacks aim to destabilize systems dynamics by intercepting communication (DoS attacks) or injecting false data (false-data injection attacks), our future work will design more various attacks, which can achieve attack purposes and not destabilize systems. Furthermore, as our future work, suitable countermeasure strategies for these attacks will be studied.

References

[1] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 427–438, Feb. 2013.

[2] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015.

[3] Y. Lee, J. Trevathan, I. Atkinson, and W. Read, "An intelligent agent system for managing heterogeneous sensors in dispersed and disparate wireless sensor network," *Int. J. Sens. Netw.*, vol. 27, no. 3, pp. 149–162, 2018.

[4] Y. Yang, Y. Xiao, and T. Li, "A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 815–841, 2nd Quart., 2021.

[5] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, Sep. 2016.

[6] M. Valero, F. Li, and W. Song, "Smart seismic network for shallow subsurface imaging and infrastructure security," *Int. J. Sens. Netw.*, vol. 31, no. 1, pp. 10–23, 2019.

[7] J. Gao *et al.*, "SCADA communication and security issues," *Security Commun. Netw.*, vol. 7, no. 1, pp. 175–194, 2014.

[8] H. J. LeBlanc and X. D. Koutsoukos, "Consensus in networked multi-agent systems with adversaries," in *Proc. 14th Int. Conf. Hybrid Syst. Comput. Control*, Chicago, IL, USA, Apr. 2011, pp. 281–290.

[9] I. Shames, A. M. H. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed fault detection for interconnected second-order systems," *Automatica*, vol. 47, no. 12, pp. 2757–2764, 2011.

[10] Z. Feng, G. Hu, and G. Wen, "Distributed consensus tracking for multi-agent systems under two types of attacks," *Int. J. Robust Nonlinear Control*, vol. 26, no. 5, pp. 896–918, 2016.

[11] Z. Feng, G. Wen, and G. Hu, "Distributed secure coordinated control for multiagent systems under strategic attacks," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1273–1284, May 2017.

[12] W. He, X. Gao, W. Zhong, and F. Qian, "Secure impulsive synchronization control of multi-agent systems under deception attacks," *Inf. Sci.*, vol. 459, pp. 354–368, Aug. 2018.

[13] D. Ding, Z. Wang, D. W. C. Ho, and G. Wei, "Observer-based event-triggering consensus control for multiagent systems with lossy sensors and cyber-attacks," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 1936–1947, Aug. 2017.

[14] M. Zhu and S. Martinez, "On the performance analysis of resilient networked control systems under replay attacks," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 804–808, Mar. 2014.

[15] X.-M. Li, Q. Zhou, P. Li, H. Li, and R. Lu, "Event-triggered consensus control for multi-agent systems against false data-injection attacks," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 1856–1866, May 2020.

[16] X.-M. Zhang, Q.-L. Han, X. Ge, and L. Ding, "Resilient control design based on a sampled-data model for a class of networked control systems under denial-of-service attacks," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3616–3626, Aug. 2020.

[17] C. Peng, J. Li, and M. Fei, "Resilient event-triggering h_∞ load frequency control for multi-area power systems with energy-limited dos attacks," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 4110–4118, Sep. 2017.

[18] C. Peng and H. Sun, "Switching-like event-triggered control for networked control systems under malicious denial of service attacks," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3094–3103, Sep. 2020.

[19] X. Shao and D. Ye, "Fuzzy adaptive event-triggered secure control for stochastic nonlinear high-order mass subject to DoS attacks and actuator faults," *IEEE Trans. Fuzzy Syst.*, early access, Oct. 5, 2020, doi: 10.1109/TFUZZ.2020.3028657.

[20] T.-Y. Zhang and D. Ye, "Distributed secure control against denial-of-service attacks in cyber-physical systems based on K-connected communication topology," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3094–3103, Jul. 2020.

[21] X.-M. Zhang *et al.*, "Networked control systems: A survey of trends and techniques," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 1, pp. 1–17, Jan. 2020.

[22] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures," *Rel. Eng. Syst. Safety*, vol. 96, no. 1, pp. 11–25, 2011.

[23] J. Liu, Y. Xiao, S. Li, W. Liang, and C. L. P. Chen, "Cyber security and privacy issues in smart grids," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 981–997, 4th Quart., 2012.

[24] J. Jow, Y. Xiao, and W. Han, "A survey of intrusion detection systems in smart grid," *Int. J. Sens. Netw.*, vol. 23, no. 3, pp. 170–186, 2017.

[25] T. Mahjabin, Y. Xiao, T. Li, and C. L. P. Chen, "Load distributed and benign-bot mitigation methods for IoT DNS flood attacks," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 986–1000, Feb. 2020.

[26] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. Robot. Autom.*, vol. 19, no. 6, pp. 933–941, Dec. 2003.

[27] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Auton. Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[28] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: Theories and applications," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 340–348, Jan. 2015.

[29] X. Wang, V. Yadav, and S. N. Balakrishnan, "Cooperative UAV formation flying with obstacle/collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 4, pp. 672–679, Jul. 2007.

[30] J. Wang and M. Xin, "Integrated optimal formation control of multiple unmanned aerial vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 5, pp. 1731–1744, Sep. 2013.

[31] Z. Yan, X. Pan, Z. Yang, and L. Yue, "Formation control of leader-following multi-UUVs with uncertain factors and time-varying delays," *IEEE Access*, vol. 7, pp. 118792–118805, 2019.

[32] Z. Yan, Z. Yang, L. Yue, L. Wang, H. Jia, and J. Zhou, "Discrete-time coordinated control of leader-following multiple AUVs under switching topologies and communication delays," *Ocean Eng.*, vol. 172, pp. 361–372, Jan. 2019.

[33] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2027–2051, 3rd Quart., 2016.

[34] K. Haataja and P. Toivanen, "Two practical man-in-the-middle attacks on Bluetooth secure simple pairing and countermeasures," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 384–392, Jan. 2010.

[35] M. Agarwal, S. Biswas, and S. Nandi, "Advanced stealth man-in-the-middle attack in WPA2 encrypted Wi-Fi networks," *IEEE Commun. Lett.*, vol. 19, no. 4, pp. 581–584, Apr. 2015.

[36] M. Oh, Y.-G. Kim, S. Hong, and S. Cha, "ASA: Agent-based secure ARP cache management," *IET Commun.*, vol. 6, no. 7, pp. 685–693, 2012.

[37] J. Deng, K. Meng, Y. Xiao, and R. Xu, "Implementation of DoS attack and mitigation strategies in IEEE 802.11b/g WLAN," in *Proc. Sens. Command Control Commun. Intell. (C3I) Technol. Homeland Security Homeland Defense IX*, vol. 7666. Orlando, FL, USA, Apr. 2010, pp. 29–38.

[38] Z. Li, T. Jing, Y. Huo, and J. Qian, "Achieving secure communications in multi-antenna cooperative cognitive radio networks using cooperative jamming," *Int. J. Sens. Netw.*, vol. 22, no. 2, pp. 100–110, 2016.

[39] Z. Lin, L. Wang, Z. Han, and M. Fu, "Distributed formation control of multi-agent systems using complex Laplacian," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1765–1777, Jul. 2014.

[40] X. Liu, S. S. Ge, C.-H. Goh, and Y. Li, "Event-triggered coordination for formation tracking control in constrained space with limited communication," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1000–1011, Mar. 2019.

[41] M. Cao, F. Xiao, and L. Wang, "Event-based second-order consensus control for multi-agent systems via synchronous periodic event detection," *IEEE Trans. Autom. Control*, vol. 60, no. 9, pp. 2452–2457, Sep. 2015.

Yue Yang (Student Member, IEEE) received the bachelor's degree in nautical science from Dalian Maritime University, Dalian, China, in 2014, where he is currently pursuing the Ph.D. degree in traffic information technology and control.

His current research interests include cybersecurity, multiagent systems, and formation control.

Yang Xiao (Fellow, IEEE) received the B.S. and M.S. degrees in computational mathematics from Jilin University, Changchun, China, in 1989 and 1991, respectively, and the M.S. and Ph.D. degrees in computer science and engineering from Wright State University, Dayton, OH, USA, in 2000 and 2001, respectively.

He is currently a Full Professor with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA. He has published over 300 SCI-indexed journal papers (including over 50 IEEE/ACM TRANSACTIONS papers) and 250 EI indexed refereed conference papers related to these research areas. His current research interests include cyber-physical systems, the Internet of Things, security, wireless networks, smart grid, and telemedicine.

Prof. Xiao currently serves as the Editor-in-Chief of *Cyber-Physical Systems*. He has served on editorial board or Associate Editor of 20 international journals, including the IEEE TRANSACTIONS ON CYBERNETICS in 2020, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS from 2014 to 2015, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2007 to 2009, and IEEE COMMUNICATIONS SURVEY AND TUTORIALS from 2007 to 2014, a Guest Editor over 20 times of different international journals, including the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE NETWORK, IEEE WIRELESS COMMUNICATIONS, and *ACM/Springer Mobile Networks and Applications*. He was a Voting Member of the IEEE 802.11 Working Group from 2001 to 2004, involving the IEEE 802.11 (WIFI) standardization work. He is an IET Fellow.

Tieshan Li (Senior Member, IEEE) received the B.S. degree in ocean fisheries engineering from the Ocean University of China, Qingdao, China, in 1992, and the Ph.D. degree in vehicle operation engineering from Dalian Maritime University (DMU), Dalian, China, in 2005.

He was a Lecturer with DMU from 2005 to 2006, an Associate Professor from 2006 to 2011, and he has been a Ph.D. supervisor since 2009, where he has been a Full Professor since 2011 and the Chaired Professor since 2021. He is currently a Tenured Professor with the University of Electronic Science and Technology of China, Chengdu, China. From 2007 to 2010, he worked as a Postdoctoral Scholar with the School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai, China. From 2008 to 2009 and 2014 to 2015, he visited the City University of Hong Kong, Hong Kong, as a Senior Research Associate. Since 2013, he has visited the University of Macau, Macau, China, as a Visiting Scholar many times. His research interests include intelligent learning and control for nonlinear systems, multiagent systems, and their applications to marine vehicle control.