IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 30, 2019, accepted January 10, 2020, date of publication January 15, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966764

Data Logic Attack on Heavy-Duty Industrial Manipulators

LIANPENG LI[1], LUN XIE[1], BING HAO[2], LIUSONG YANG[2], TONGHAI HU[2], AND ZHILIANG WANG[1]

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2]CITIC Heavy Industries Company, Ltd., Luoyang 471039, China

Corresponding author: Lun Xie (xielun@ustb.edu.cn)

This work was supported in part by the National Key Research and Development Program under Grant 2017YFB1302104, and in part by the National Natural Science Foundation of China under Grant 61672093 and Grant 61432004.

ABSTRACT The anticipated widespread use of the heavy-duty industrial manipulator (HDIM) makes it an important role in the field of modern industrial automation. Research on the attack of cyber-physical systems based on industrial manipulator vulnerabilities is booming, while there are few studies on the data logic and attack impact for HDIMs. This paper proposes a new cyber-physical attack mechanism named data logic attack mechanism on HDIMs, including network protocol data logic attack, system data integrity logic attack, and process logic attack. Meanwhile, data logic attack models for HDIMs and an attack impact analysis model are established. Besides, for the proposed data logic attack mechanism, a hardware-in-the-loop cosimulation based on Simulink and Adams is carried out to demonstrate the impact of data logic attacks on the system integrity, availability, accuracy, and integrity. A test platform has also been established to test the attack mechanism's effectiveness. The results of cosimulation and test show the attack impact ranking and effectiveness of the attack mechanism.

INDEX TERMS Industrial manipulator, data logic attack, impact analysis model, cosimulation.

I. INTRODUCTION

The heavy-duty industrial manipulator (HDIM) is one type of industrial manipulators. It is a multi-degree-of-freedom (DOF) manipulator for the industrial field, and can automatically perform operations and realize various functions by its power and control ability [1]. It has been used in production processes(e.g., equipment manufacturing, welding inspection, and fine assembly), and widely applied in many industries such as construction machinery, rail transportation, electric power, medicine, and mechanization in mining [2], [3]. Once the security of HDIMs is destroyed, the impact is enormous due to the complex working environment of the heavy-duty manipulator, the massive weight of the mechanical structure, and the weak safety mechanism.

At present, industrial manipulators face serious security challenges [4]. For example, as of March 2017, at least 83600 industrial manipulators had been attacked by the Internet through File Transfer Protocol (FTP) servers or unprotected routers[5]. These challenges are mainly reflected in the vulnerability of industrial manipulators in cyber-physical systems (CPSs). The vulnerabilities mainly include the following three aspects:

1) Network communication vulnerability. Many industrial manipulators have the functions of remote monitoring, operation, and maintenance [6], which bring convenience and increase the risk of their intrusion from the network. If the controller does not enforce end-to-end program integrity, an attacker can leverage a file system or an authentication-bypass vulnerability to alter a communication logic arbitrarily.
2) System data vulnerability. Many manipulators run under open source code and share the similar architectures, component types, and functional features [7]. Attackers can easily obtain a robotic controller structure, exploit vulnerabilities, and create targeted cyber-physical attacks on data by reading free technical documentation or other available information from a network. Meanwhile, the weak data protection capabilities and security mechanisms of industrial manipulators have been proved by Trend Micro [8].
3) Physical process vulnerability. As devices move from hardwired logic to more flexible software-based

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan M. Abu-Mahfouz.

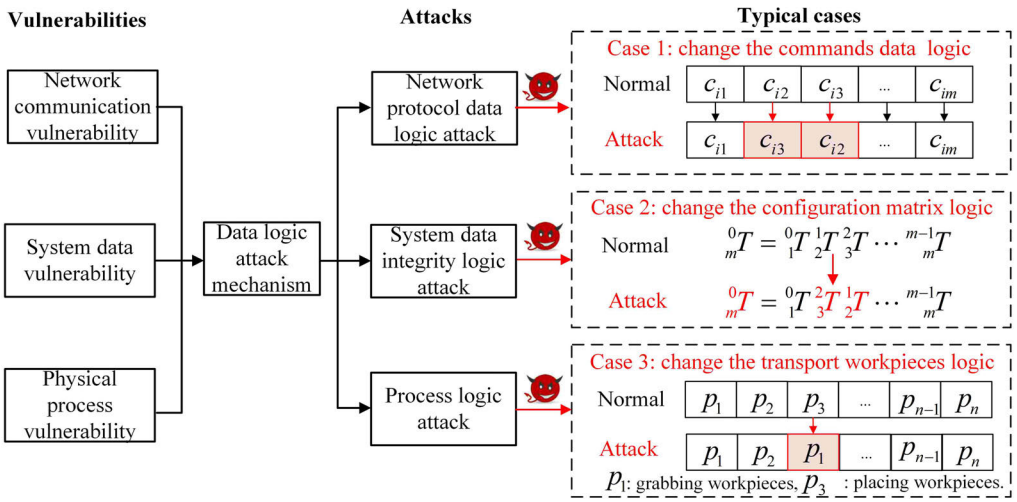


FIGURE 1. Vulnerabilities, mechanism, and typical attack cases on heavy-duty industrial manipulators (HDIMs).

implementations, the security issues are increased. Moreover, the physical process logic has not attracted enough attention to industrial manipulators [9]. Tampering of production logic will put industrial manipulators at risk.

For the vulnerabilities of industrial manipulators, there have been some related attack researches, such as the denial of service (DOS) attacks [10], data integrity attacks [11], and spoofing attacks [12]. However, these studies focus on obtaining system information, generating malicious traffic, and deceiving system devices, and lack the analyses of industrial manipulators data logic. An attacker can affect the cyber-physical state of an industrial manipulator by destroying the data logic, such as tampering the protocol data logic, modifying control parameters logic, and disrupting the production process logic.

The purpose of this paper is to analyze the data logic attack problems of HDIMs. To this end, a data logic attack mechanism is proposed. As shown in Figure 1, the mechanism consists of three data logic attacks: network protocol data logic attack, system data integrity logic attack, and process logic attack.

1) The network protocol data logic attack usually tampers with data packet logic through protocol and attacks industrial manipulators. Destroying the communication protocol, reordering, replaying, and dropping packets all destroy the protocol logic. Case 1 in Figure 1 is launching the attack by reordering the command packets, which destroys the communication logic. $c_{i1}, c_{i2}, c_{i3}, \cdots, c_{im}$ are a series of control commands to the corresponding actuators, and m is the number of actuators (the joints of an HDIM), $m > 3$. The exchange of c_{i2} and c_{i3} destroys the command logic, causing the wrong commands to be assigned to the correct actuators to generate data logic attacks.
2) The system data integrity logic attack includes modifying control algorithms logic and tampering with model configuration parameters to destroy system data logic in HDIMs, for example, tampering with some parameters will cause the industrial manipulators to go beyond the working range to form an attack. A case provided in case 2 (shown in Figure 1) illustrates the system data integrity logic attack by intentionally manipulating the kinematics matrices. ${}^{l-1}_{l}T$ is a configuration matrix of the kinematics [13], $1 \leq l \leq m$.
Modifying the logic of the matrix ${}^{0}_{m}T$ will result in logic attacks on system data.
3) The process logic attack. Commands are in accordance with the protocol specification, but violate the production logic process of the HDIMs, making the system in a dangerous state (e.g., the order in which the manipulator grabs, transports and places the workpiece). In case 3 shown in Figure 1, p_s represents a physical process of transporting workpieces, $1 \leq s \leq n$, $n > 4$. In the process of placing workpieces, the actuator executed the grabbing workpiece commands when the workpiece needed to be placed, which will cause a process data logic attack.

In general, we focus on the vulnerabilities of HDIMs and propose a data logic attack mechanism and an impact analysis model. The main contributions of this work are summarized as follows:

1) The vulnerabilities of industrial manipulators are analyzed in three aspects. Furthermore, the data logic attack mechanism is proposed based on the established HDIM system model. The attack mechanism consists of network protocol data logic attack, system data integrity logic attack, and process logic attack, which are all described and modeled.
2) The impact analysis model for the data logic attack mechanism is presented based on system integrity,

system availability, system accuracy, and system security. Through model calculation, the impact analysis model can describe the impact of data logic attacks digitally.

3) The hardware-in-the-loop cosimulation based on Matlab and Adams has been established, eight attack scenarios about the three data logic attacks are modeled and simulated. Moreover, the physical test platform is built. The impact of this attack mechanism on HDIM is analyzed based on the simulation and test. The results show the attack impact ranking and effectiveness of the attack mechanism.

The rest of the paper is organized as follows. Section II describes the related work. Section III shows the data logic attack mechanism and impact analysis model. In Section IV, a cosimulation and test platform are established to test the attack mechanism. Finally, conclusions and future work are summarized in Section V.

II. RELATED WORK

Recently, the cyber-physical security of industrial manipulators has arisen much attention. In this work, the vulnerabilities have been described in three aspects: network communication vulnerability, system data vulnerability, and physical process vulnerability. Researchers have done much work on these vulnerabilities.

A. NETWORK COMMUNICATION

In terms of network communication, DOS attacks are widely researched and applied [14]. Hu *et al.* [15] discussed the periodic DOS attacks on network communication and proposed a criterion to characterize the DOS parameters quantitatively. Reference [16] proposed a multi-manipulator motion planning model for missions captured in the presence of DOS attacks against the communication between robots and base stations. Meanwhile, machine learning is widely used in the cyber-domain attacks. Reference [17] analyzed the issues in network communication and proposed a cyber-attack against autonomous robot machine learning strategies. Poisoning attacks can occur during the training phase on training data in the form of either injection or modification. Biggio *et al.* [18] considered poisoning attacks on the application of the intrusion detection system and demonstrated that injecting the malicious sample into the training set by network communication will reduce the performance of the algorithm. Ahmad *et al*. [19] developed a robot attack tool (RAT) for implementing cybersecurity attacks to attack network communication, resulting in robot DOS and non-response to mobile commands.

For the network communication vulnerability, the current attack research focuses on DOS attacks and launching attacks based on machine learning, involving few data logic attacks on network domains.

B. SYSTEM DATA

The system data security is the guarantee of the safe and reliable operation of industrial manipulators[20]. During a system data transmission process, attackers modified the state of the exchange between the master and slave manipulators by introducing a static malicious content modification attack, which made the joint speed of the manipulator unstable and realized the content modification attack of the bilateral teleoperation system [21]. Sabaliauskaite *et al.* [22] implemented three stealth attacks on the Amiobobot robot: surge, offset, and geometric attacks, which broke the integrity of the system data. Destroying the state data measured or generated by the robot produces a data integrity attack, which can cause mispredictions in the operation or planning, and lead to reduce planning efficiency and security incidents [23]. Bonaci *et al.* [24] studied the impact of data integrity and recovery tasks on attacked robots and pointed out that the current robot system data is not strictly protected. Besides, the team's Alemzadeh *et al.* [25] analyzed the security threats of Raven II robots used in surgery. These attacks exploit the vulnerabilities of the robot system and inject malicious control commands to the robot by inferring the critical time during a surgery, leading to catastrophic consequences in the physical system.

For the system data vulnerability, the current attack research mainly launches attacks by destroying data integrity. But few studies consider the purposeful destruction of data logic for industrial manipulators, even if they achieved a similar attack.

C. PHYSICAL PROCESS

As for physical processes, the security of industrial manipulator controllers was systematically analyzed, and a production process tampering attack method was proposed. However, there are no data logic analyses or physical tests on the impact of attacks [26]. Li *et al*. [27] proposed a sequential logic attack based on SCADA systems and simulated it under Simulink to analyze the possible physical effects. Quarta *et al.* [28] proposed against control logic could result in damage to its physical parts, simply destroying the workpiece or surrounding environment. Similarly, there is a less further study from the perspective of the data logic model. Monteuuis *et al.* [29] proposed an attack for robot data integrity, describing in detail the attacks occurring in the real environment and theoretically analyzing the possible impact of the attack without physical tests. Pogliani *et al*. [30] explored specific attack vectors. By using these attack vectors, the interaction between the robot and its physical environment can be broken, violating the basic operational requirements of the robot. For example, an attacker could insert small defects, modify a workpiece, or fully compromise a company's manufacturing process.

The physical process vulnerability has been utilized to build cyber-physical attacks. While these studies lack systematic theoretical analyses for data logic and the research results are basically in the stage of theory and simulation. It can be seen that once the data logic is destroyed, it is likely to cause huge damage to industrial manipulators.

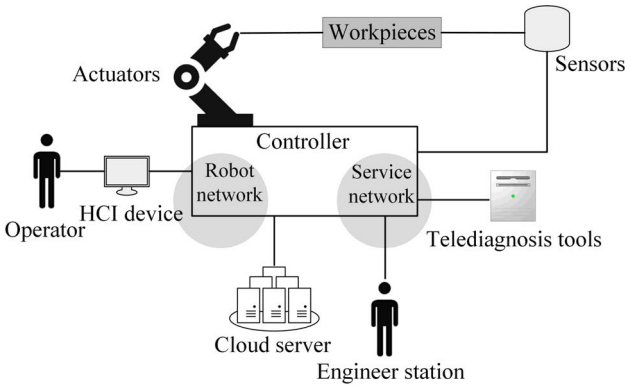


FIGURE 2. A heavy-duty industrial manipulator (HDIM) system.

To sum up, for the three vulnerabilities mentioned above for industrial manipulators, existing research has focused on DOS attacks and destroying the system data integrity. As far as we know, there is less research on the data logic attack on HDIMs. Motived by this, the research has been carried out.

III. MODELING DATA LOGIC ATTACK MECHANISM OF HDIMS

The HDIM is a multi-input, multi-output, highly nonlinear, and strongly coupled integrated intelligent system, and one of the most complex CPSs [31].

As shown in Figure 2, a typical HDIM system usually consists of a human-computer interaction (HCI) device, an engineering station, telediagnosis tools, a control loop, and communicates using an industrial network protocol [32].

The HCI device converts the operator's operation commands into the end position and attitude of the manipulator based on the force interaction and the visual interaction and realizes precise interactive control operation. The engineering station is used to configure control algorithms and adjust control parameters. Telediagnosis tools are used to prevent, identify, and recover abnormal conditions for basic system maintenance. The control loop mainly includes sensors, controllers, and actuators. As for the industrial bus, it is the control network protocol for communication. Based on the control algorithms and the sensing data read from sensors, the controller generates control commands, which will be transmitted to actuators. Then, the actuators respond to the control command data, and the response data are again transmitted by sensors to the controller. A secure HDIM system should read the precise values from sensors and send the correct and accurate commands to the actuators so that the movements are performed under control.

A. SYSTEM MODEL

The research object of this work is a six-degrees-of-freedom (6-DOFs) HDIM.

As shown in Figure 3, the mechanical composition of the HDIM consists of six joints and an end gripper.

Based on the mechanical configuration of the HDIM, the 6-DOFs Cartesian coordinate systems are established,

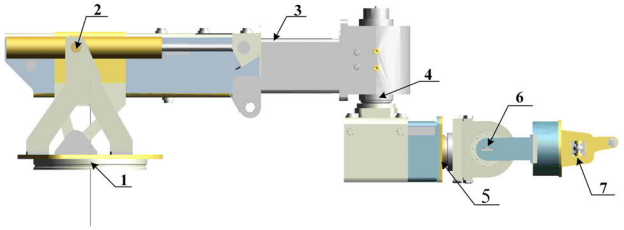


FIGURE 3. The structure of the six-degrees-of-freedom (6-DOFs) HDIM. 1: Rotary joint. 2: Pitch joint 1. 3: Telescopic joint. 4: Swing joint. 5: Roll joint. 6: Pitch joint 2. 7: End gripper.

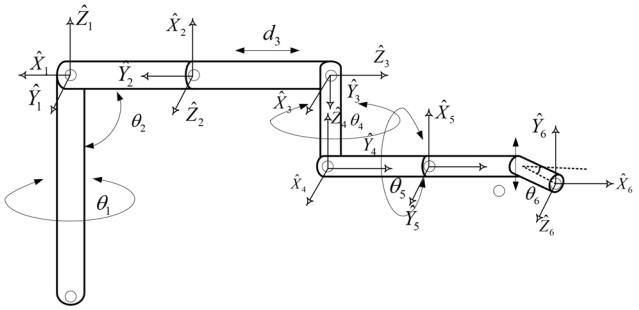


FIGURE 4. The 6-DOFs Cartesian coordinate systems.

TABLE 1. The D-H parameters.

i	Twist angle α_{i-1}	Link length a_{i-1}	Link offset d_i	Joint angle θ_i
1	0	0	0	θ_1
2	-90°	$L_2 = 2320$	0	$\theta_2 - 90^\circ$
3	90°	0	d_3	90°
4	90°	0	$L_4 = 511$	θ_4
5	-90°	0	$L_5 = 469$	$\theta_5 - 90^\circ$
6	-90°	$L_6 = 180$	0	$\theta_6 - 90^\circ$

as shown in Figure 4. The coordinate systems range from coordinate system {0} to coordinate system {6}.

In addition, according to the coordinate systems, the Denavit-Hartenberg (D-H) parameter information of the HDIM can be obtained in Table 1 [33].

Table 2 gives the notations and descriptions.

Then, we established the HDIM Kinematic Model based on the above work in Algorithm 1.

Where, (1), as shown at the bottom of the next page and

$$R = \begin{cases} -180^\circ \le j_{i_1} \le 180^\circ \\ -35^\circ \le j_{i2} \le 40^\circ \\ 0 \le j_{i3} \le 600mm \\ -35^\circ \le j_{i4} \le 95^\circ \\ 65^\circ \le j_{i5} \le 65^\circ \\ -105^\circ \le j_{i6} \le 25^\circ, \end{cases} \tag{2}$$

TABLE 2. Notations and descriptions.

Notations	Descriptions
s_i	System state
o_i	Output of sensors
c_i	Control command
A_s, D_s, C_e, D_e	System configuration matrices
B_s	Positive kinematics matrix
H	A series of HCI commands
G	Configuration information
R	Motion range of joints
$joint_i$	Controlled joint
h_i	HCI command
d_i	Motion error of $joint_i$
μ_i	Sensor noise
φ	Allowed packet loss rate
w_i	System noise

The model of the HDIM system is established by considering the control commands, sensor feedback, and system noise:

$$\begin{cases} s_{i+1} = A_s(s_i + \Delta s_i) + B_s(c_i + d_i) + D_s w_i \\ e_i = C_e(s_i + \Delta s_i) + D_e \mu_i. \end{cases} \tag{3}$$

Here, B_s is the positive kinematics matrix, in this system, it is ${}^0_m T$. e_i denotes the end joint's position and attitude data measured by sensors.

B. DATA LOGIC ATTACKS MODEL

Based on the HDIM system model, this part modeled the data logic attack mechanism, this part modeled the data logic attack mechanism, whose key is to destroy the data logic in cyber and physical state. The models include the network protocol data logic attack model, the system data integrity logic attack model, and the process logic attack model. As shown in Figure 5, the more specific system descriptions and nodes that may be attacked by the attack mechanism are presented based on the HDIM system structure, system model, and interface information.

1) MODEL OF NETWORK PROTOCOL DATA LOGIC ATTACK

The controller obtains control commands c_i based on the configuration information G, pre-written data logic L (the logic of Algorithm 1), and outputs of sensors o_i (including the end joint pose e_i and six joint displacements j_i of HDIMs). Here, $c_i = (c_{i1}, c_{i2}, c_{i3}, \cdots, c_{im})$, $j_i = (j_{i1}, j_{i2}, j_{i3}, \cdots, j_{im})$. In this work, the data logic in network protocol mainly refers to the logic of data packets from the controller to actuators and from the sensors to the controller. Thus, the network protocol data logic attack is deployed in the communication network between the central controller and the actuators or sensors.

The process of network protocol data logic attack is shown in Figure 6.

We mainly considered tampering with the control command packet logic and modeled this attack.

The attack model is

$$\begin{cases} s_{i+1} = A_s(s_i + \Delta s_i) + B_s(\tilde{c}_i + d_i) + D_s w_i \\ e_i = C_e(s_i + \Delta s_i) + D_e \mu_i, \end{cases} \tag{4}$$

where $\tilde{c}_i$ is the attacked command data, and the data logic has been destroyed by reordering data packets, delaying data packets, or dropping data packets. The corresponding attacks have been modeled in IV. B. $\tilde{o}_i$ is the sensor data with the destroyed logic, which will also form an attack on the system status of HDIMs.

2) MODEL OF SYSTEM DATA INTEGRITY LOGIC ATTACK

Data integrity logic attacks can destroy the system data logic and manipulate the manipulator's behaviors. In this paper,

Algorithm 1 The HDIM Kinematic Model

Inputs System state s_i ←{end position p_i^d and attitude r_i^a}
Initialization D-H parameters in Table 1 and s_{i-1}
Outputs Joint displacements $j_i = (j_{i1}, j_{i2}, \ldots, j_{im})$
1: for i from 1 to m do
2: Calculate the joint transformation matrix from coordinate system $\{i\}$ to coordinate system $\{i-1\}$: ${}^{i-1}_i T$
3: end for
4: Calculate the joint transformation matrix from the terminal coordinate system $\{m\}$ to the initial coordinate system $\{0\}$: ${}^0_m T = {}^0_1 T {}^1_2 T {}^2_3 T \cdots {}^{m-1}_m T$
5: Obtain the end position and attitude matrix T_i
6: Computer the joint displacement j_i with respect to s_i by $T_i = {}^0_m T$
7: if j_i within R ← the motion range of joints then
8: System state s_i exists
7: Obtain the kinematic model $s_i = s_{i-1} + T_i j_i$

$$ {}^{i-1}_i T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i \cos\alpha_{i-1} & \cos\theta_i \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1} d_i \\ \sin\theta_i \sin\alpha_{i-1} & \cos\theta_i \sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1, \end{bmatrix} \tag{1}$$

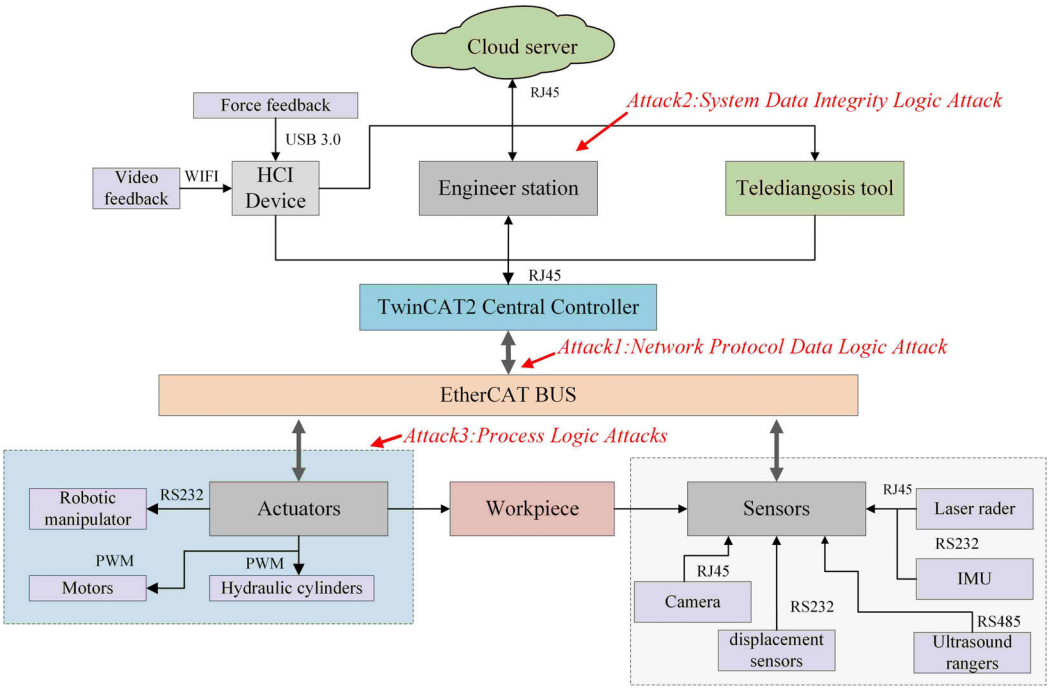


FIGURE 5. The data logic attack mechanism.

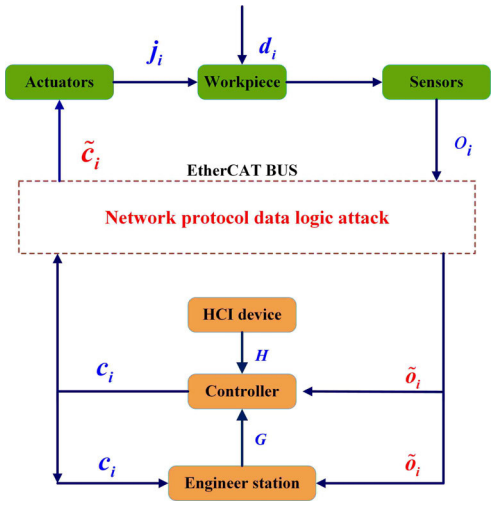


FIGURE 6. Network protocol data logic attack.

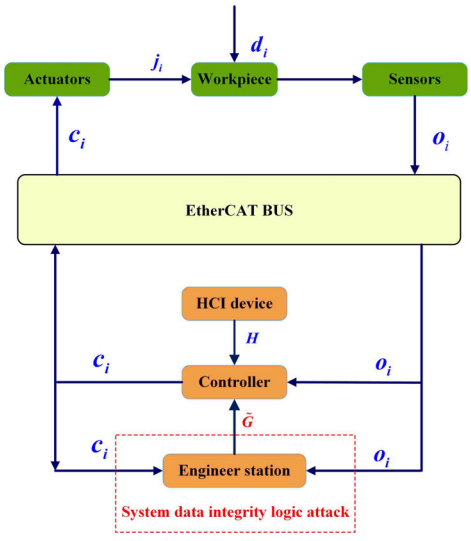


FIGURE 7. System data integrity logic attack.

system data integrity logic attacks mainly include the following two types: (1) manipulate the data or software logic on the controller, and (2) manipulate the data or software logic on the sensors. At this point, we introduced the modification of the system configurations to achieve the attack on the controller. The configuration information G consists of model configuration matrices such as B_s, pre-written data logic L, and parameters such as motion range of joints R. Figure 7 shows the system data integrity logic attack.

We modeled the attack model by tempering the data logic of configuration matrices.

$$\begin{cases} s_{i+1} = \tilde{A}_s(s_i + \Delta s_i) + \tilde{B}_s(c_i + d_i) + \tilde{D}_s w_i \\ e_i = \tilde{C}_e(s_i + \Delta s_i) + \tilde{D}_e \mu_i \end{cases} \tag{5}$$

Here, $\tilde{A}_s$, $\tilde{B}_s$, $\tilde{D}_s$, $\tilde{C}_e$ and $\tilde{D}_e$ are the tampered configuration matrices. The data logic of these matrices has been destroyed, like Case 2 in Figure 1.

The data integrity logic of the HDIM system will be compromised when one of the configuration parameters or the logic of algorithms is simultaneously maliciously or purposefully tampered with and then written to the controller. Two system data integrity logic attacks have been modeled later in cosimulation to test the attack impact.

3) MODEL OF PROCESS LOGIC ATTACK

The ultimate target of the process logic attack is to disrupt the operation logic of actuators, and the attack is usually

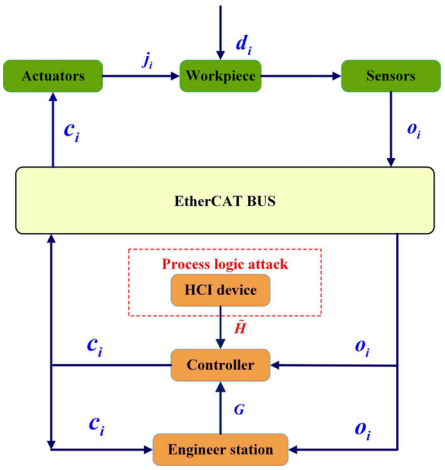


FIGURE 8. Process logic attack.

implemented by attacking the HCI device and sending HCI commands randomly or purposefully. It is easy to invade the device by observing and analyzing the commands for attackers [34]. The HCI device has a higher priority than the motions planned by the controller, and the commands from the HCI device will be carried out first. During a job, the attacked HCI device implements a process logic attack, randomly issuing grab, transport, and placement commands, which may cause serious physical damage. The process logic attack is shown in Figure 8.

The attack model is

$$\begin{cases} s_{i+1} = A_s(s_i + \Delta s_i) + B_s(\tilde{h}_i + d_i) + D_s w_i \\ e_i = C_e(s_i + \Delta s_i) + D_e \mu_i, \end{cases} \tag{6}$$

where h_i represents the HCI commands issued by an HCI device, $\tilde{h}_i$ is the attacked HCI commands. We list three kinds of process logic attacks of HDIMs. (1) The placement command is issued before the process of grabbing the workpiece, or the transport instruction is issued without grabbing the workpiece. (2) The placement command is issued before reaching the installation position. (3) The transport command is continued to be sent when the manipulator has reached the installation position.

Through the information collection, network scanning, and cracking accounts, an attack program with the data logic attack mechanism can be implanted to store, modify, and destroy the data logic of HDIMs, and realize the three attacks mentioned above.

C. IMPACT ANALYSIS MODEL

To analyze the impact of the proposed data logic attack mechanism, we mainly consider the system integrity, system availability, system accuracy, and system security.

1) System integrity (IN) includes transmitted data integrity (TR) and system configuration integrity (CI).
2) System availability (AV) mainly includes hardware availability (HA), service availability (SA), and link availability (LA).
3) System accuracy (AC) mainly refers to the degree of disturbance (DI) and task completion degree (CI) of the attack relative to normal physical processes.
4) System security (SE) refers to the degree of damage (DA) of the attack to the system and whether there exists a threat (ET) to the safety of the operators.

I, Φ, Ω, and Ψ are used to indicate IN, AV, AC, and SE. Accurately, I_i^j represents the integrity impact of the attack Att_i on the jth observation, $j = 1, 2$. I_i^1 denotes the impact on TR, I_i^2 denotes the impact on CI. Similarly, Φ_i^k means the availability impact of the attack Att_i on the kth observation, $k = 1, 2, 3$. Φ_i^1, Φ_i^2, and Φ_i^3 respectively represent the impact on HA, SA, and LA. Ω_i^m indicates the accuracy impact of the attack Att_i on the mth observation, $m = 1,\ 2,\ 3$. Ω_i^1 and Ω_i^2 respectively represent the impact on DI to the system position and DI to the system attitude. Ω_i^3 takes the CI of the operation, and mainly considers the ratio of the difference between non-attacked position and attacked position. Ψ_i^u indicates the DA to the system by the attack Att_i, $u = 1, 2, 3, 4, 5, 6$. Here, it mainly means that the force of the uth joint of HDIM exceeds the degree of joint limitation. Ψ_i^t denotes the ET to the operators.

I_i^j, Φ_i^k, Ω_i^m, and Ψ_i^l are calculated as follows under the same conditions.

Firstly, for the integrity of the observation, we proposed the rule I. $\mathrm{I}_i^j \in \{0, 1\}$, this means that once the attack Att_i has an impact on the integrity of any observations, I_i^j takes 1, otherwise takes 0. For TR, the desired transmission data d_d and the actual received data d_a are compared, and the ratio of the difference between the two and d_d is recorded as f_d. If f_d is greater than the allowed packet loss rate of the system, then I_i^j takes 1, otherwise, it takes 0. For the impact of CI, it mainly considers whether the logic of system configuration matrices A_s, B_s, C_e, D_s, and D_e have been destroyed. In case of change, I_i^j takes 1, otherwise, it takes 0. Then,

$$g(\mathrm{I}_i) = \sum_{j=1}^{2} \mathrm{I}_i^j. \tag{7}$$

Secondly, the rule Φ was proposed for the impact of AV. $\Phi_i^k \in \{0, 1\}$, this indicates that if the attack Att_i has an impact on the integrity of the kth observation, it is 1, otherwise, it is 0. For HA, the judgment standard mainly refers to whether the hardware such as the manipulator joint motor and controller alarms. The triggering of the alarm includes three aspects: (1) determining whether the structural deformation of the joint arm is within the safe range based on the displacement sensor, (2) judging whether it is within the safe range based on the motor output force and torque, and (3) determining whether the processor temperature, memory, and drive are faulty based on the data of the controller. If an alarm occurs, Φ_i^k takes 1, otherwise, Φ_i^k takes 0. For SA, it mainly judges whether the engineer station, controllers, actuators, and sensors provide service. If one of them is unable to complete its task, Φ_i^k takes 1. At the same time, we mainly test whether normal communication is available among the components

for LA, and the third-party tool Wireshark is needed to judge the communication link [35].

$$g(\Phi_i) = \sum_{k=1}^{3} \Phi_i^k \tag{8}$$

$$\Omega_i^1(t) = \begin{cases} \dfrac{\left|p_{i\_\max}^{da} - p_{i\_\min}^{da}\right|}{\left|p_{i\_\max}^{d} - p_{i\_\min}^{d}\right|}, & \left|p_{i\_\max}^{da} - p_{i\_\min}^{da}\right| > \left|p_{i\_\max}^{d} - p_{i\_\min}^{d}\right| \\ 0, & \left|p_{i\_\max}^{da} - p_{i\_\min}^{da}\right| \le \left|p_{i\_\max}^{d} - p_{i\_\min}^{d}\right| \end{cases}, \tag{9}$$

where p_i^d is the end position without attack, p_i^{da} is the actual position of the manipulator system in the presence of the attack Att_i, and P_i^s is the starting position of the motion. During the movement of the manipulator system without attack, $p_{i\_\min}^d$ and $p_{i\_\max}^d$ are the minimum displacement and the maximum displacement, respectively. Besides, $p_{i\_\min}^{da}$ and $p_{i\_\max}^{da}$ are the minimum and maximum displacements of the manipulator under attack Att_i, respectively.

$$\Omega_i^2(t) = \begin{cases} \dfrac{\left|r_{i\_\max}^{aa} - r_{i\_\min}^{aa}\right|}{\left|r_{i\_\max}^{a} - r_{i\_\min}^{a}\right|}, & \left|r_{i\_\max}^{aa} - r_{i\_\min}^{aa}\right| > \left|r_{i\_\max}^{a} - r_{i\_\min}^{a}\right| \\ 0, & \left|r_{i\_\max}^{aa} - r_{i\_\min}^{aa}\right| \le \left|r_{i\_\max}^{a} - r_{i\_\min}^{a}\right| \end{cases}, \tag{10}$$

where r_i^a is the desired attitude without attack, r_i^{aa} is the actual attitude of the manipulator system under attack Att_i. $r_{i\_\min}^a$ and $r_{i\_\max}^a$ are respectively the minimum attitude and the maximum attitude during the movement of the manipulator system without attack. $r_{i\_\min}^{aa}$ and $r_{i\_\max}^{aa}$ are the minimum attitude and the maximum attitude of the manipulator under attack Att_i.

$$\Omega_i^3(t) = \left|p_i^{da} - p_i^d\right| / \left|p_i^d - p_i^s\right| \tag{11}$$

This Ω_i^3 is determined by the motion range R, the control command, and the system model.

$$g(\Omega_i) = \sum_{m=1}^{3} \Omega_i^m \tag{12}$$

The above calculation rules are named rule Ω.

$$\Psi_i^u(t) = \begin{cases} F_{i\_\max}^{ua} / F_{i\_\max}^{u}, & F_{i\_\max}^{ua} > F_{i\_\max}^{u} \\ 0, & F_{i\_\max}^{ua} \le F_{i\_\max}^{u} \end{cases} \tag{13}$$

$$g(\Psi_i^u) = \sum_{u=1}^{6} \Psi_i^u \tag{14}$$

$F_{i\_\max}^u$ is the maximum joint force allowed by the observed joint U under the condition of no attack, and each joint has fixed parameters. $F_{i\_\max}^{ua}$ is the maximum force of the observation u in the presence of the attack Att_i. Equation (13) gives the calculation formula of Ψ_i^u. If attack Att_i exists a threat to the safety of the operators, it is scored as $\Psi_i^t = 1$, otherwise, it takes 0.

$$g(\Psi_i) = \sum_{u=1}^{6} \Psi_i^u + \Psi_i^t \tag{15}$$

The above attack impact calculation rules are named rule Ψ.

Calculating the impact score $G(Att_i)$ of the attack Att_i on the system IN, AV, AC, SE based on rule I, Φ, Ω, and Ψ. The impact analysis model is

$$G(Att_t) = \varphi_{\mathrm{I}} g(\mathrm{I}_i) + \varphi_{\Phi} g(\Phi_i) + \varphi_{\Omega} g(\Omega_i) + \varphi_{\Psi} g(\Psi_i). \tag{16}$$

φ_{I}, φ_{Φ}, φ_{Ω}, and φ_{Ψ} are coefficients of IN, AV, AC, and SE, respectively, and the corresponding weights are set to 0.2, 0.2, 0.2, and 0.4.

IV. COSIMULATION AND TEST

In this section, we mainly built a hardware-in-the-loop cosimulation based on Matlab/Simulink and Adams and carried out data logic attacks in several scenarios. There have eight modeled attacks in three cases. Based on the established impact analysis model, the impact and effect of the attack mechanism are analyzed, and then the effectiveness of the mechanism is verified by actual physical tests. We assume that the attacker is sufficiently aware of the system model and can tamper with the system configurations to launch an attack through technical means [36].

A. COSIMULATION SYSTEM

To test the effectiveness and impact of the data logic attack mechanism, we have built a hardware-in-the-loop cosimulation model based on an HCI feedback device, a control handle, and two software. As there is a data exchange between Simulink and Adams and includes hardware and software, the simulation is named cosimulation. The virtual control module and mechanical module are developed by Simulink and Adams respectively to realize the interaction simulation between the virtual control system and the virtual machine by sharing the virtual model. Besides, the HCI device, the control handle, and the virtual control system are used to realize the HCI function in operation. We tested the attack mechanism impact based on the established cosimulation system.

Hardware Description: the HCI feedback device named Xbox one is a kind of high-performance and programmable HCI handle. By defining the keys of the device, it can issue the data of position and attitude and realize the fine motion control and high-precision remote operation of the manipulator. Meanwhile, the handle can feedback on the interaction force between the manipulator and the operating environment.

Figure 9 shows the combination of the keys, the red coordinate system corresponds to the end position matrix of the HDIM, and the green coordinate system corresponds to the

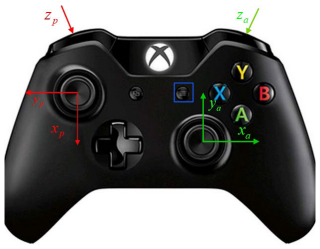


FIGURE 9. The human-computer interaction (HCI) device.

end attitude matrix. The key in the blue box is used to switch the positive and negative directions of the Z-axis. Meanwhile, the vibration frequency of left joystick is used to simulate the interaction force with the environment.

The host of simulation platform: Intel i5-7400 CPU 3.00GHz, 8G memory, and Win10 64-bit operating system.

Software Description: Matlab 2018b has more options for data analysis and simulation calculation, and runs faster than ever before. Simulink provides an integrated environment for dynamic system modeling, simulation, and comprehensive analysis. Moreover, Adams2013 is a general mechanical system dynamics simulation software that can comprehensively reflect the true characteristics of complex nonlinear systems.

The hardware-in-the-loop cosimulation system is shown in Figure 10, consisting of hardware (the HCI device and the control handle) and software (Simulink and Adams). The inputs of this system are HCI commands or preset position and attitude from data.mat, and the outputs are the pose of the end joint.

The control handle is an Xbox 360, and the interface is USB-HID. Since Simulink does not directly call the USB module, we use the converted interface USB to the serial port to achieve the communication. By defining the keys of Xbox 360, the handle is used to simulate the functions of grab and placement workpieces in the cosimulation system. Besides, it is used to switch the input of the control signal.

The HCI device adopts a serial port interface. After setting the baud rate and serial port number in Simulink, the hardware device can communicate with the virtual co-simulation system.

The data in the cosimulation are the pose data. The manipulator is the simulated actuator. The data generated during motions can be viewed and recorded in Simulink and Adams, which is similar to the function of sensors.

The control model is mainly composed of a kinematic model and a dynamic model. The control model converts the desired position and attitude into joint angular displacements, which are then converted into driving torques. The dynamic model of this HDIM is described detailedly in [37].

B. COSIMULATION MODEL

1) THE DATA LOGIC ATTACK SIMULATION

The data logic attack simulation is built on the completed hardware-in-the-loop cosimulation system. We have established the following scenarios, and eight logic attacks have been modeled and simulated.

Case 1: For network protocol data logic attacks, this case contains three types of attacks: reordering the packets, delaying data packets, and dropping data packets. Besides, we have built three attack models.

$$\tilde{c}_{io} = reorder(c_i) = (c_{i1}, c_{i2}, c_{i5}, c_{i3}, c_{i4}, c_{i6}) \tag{17}$$

In the attack function $reorder(c_i)$, the logic of control commands has been changed, and the reordered commands are allocated to the correct actuators. It will generate a data logic attack.

$$\begin{aligned}\tilde{c}_{ip} &= replay(c_i) \\ &= (c_{(i-x)1}, c_{(i-x)2}, c_{(i-x)3}, c_{(i-x)4}, c_{(i-x)5}, c_{(i-x)6}),\end{aligned} \tag{18}$$

where $x \in (0, i)$. In Equation (18), the control instructions are replayed, while retaining their legal format and syntax, without changing the performance of the target program. The old correct logic data will also destroy the network protocol data logic.

$$\tilde{c}_{ic} = drop(c_i) = (c_1, c_2, \ldots, c_{i-1}, c_{i+\gamma}, \ldots, c_n), \tag{19}$$

where $1 \leq \gamma \leq 100\varphi$. φ s the allowable packet loss rate of the manipulator, but continuous multiple packet loss will destroy the data logic and cause the jamming of the manipulator movement.

In the scenario of case 1, the attack mode is implemented by adding an attack program to the link between the control model and the manipulator to store, modify, and transmit control commands to implement the above three network protocol data logic attack models [38].

Case 2: It is possible to damage the physical parts of the manipulator if the controller writes the wrong control logic or misconfigures the system model. For the logic attack of system data integrity, this case provides two attack models.

The first is to randomly or purposefully tamper the data logic of configuration matrices, and destroy the data integrity of the system.

In this case, we choose the matrix B_s. According to the preceding, it is known that B_s represents the ${}^0_mT$ and ${}^0_mT = {}^0_1T\,{}^1_2T\,{}^2_3T \cdots {}^{m-1}_mT$. Then, we destroy the data logic of B_s in red line, and the values in blue line are the attacked system data in Equation (20).

$$\begin{aligned}s_{i+1} &= A_s(s_i + \Delta s_i) + ({}^0_1T\,{}^4_5T\,{}^2_3T\,{}^3_4T\,{}^1_2T\,{}^5_6T)(c_i + d_i) + D_s w_i \\ &= A_s(s_i + \Delta s_i) + \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} (c_i + d_i) + D_s w_i\end{aligned} \tag{20}$$

Equation (2) gives the motion range of HDIM's each joint. The parameters and corresponding joints are fixed. Once this relationship is broken, there may be a data logic problem. As shown in Equation (21), the motion range of $joint_2$ and $joint_5$ has been exchanged. Obviously, this is beyond the

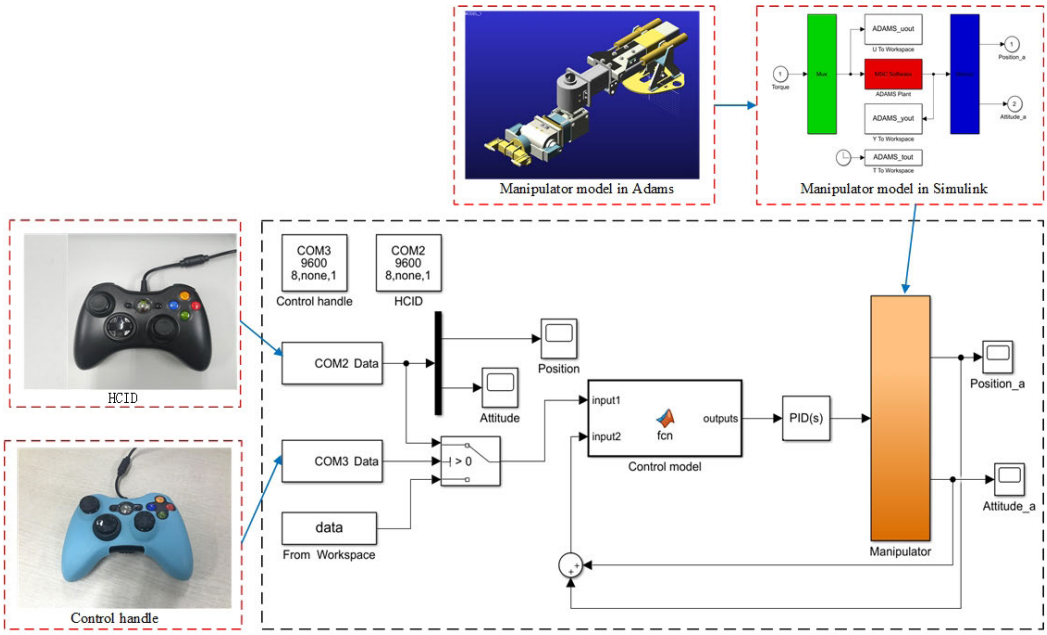


FIGURE 10. The hardware-in-the-loop cosimulation system.

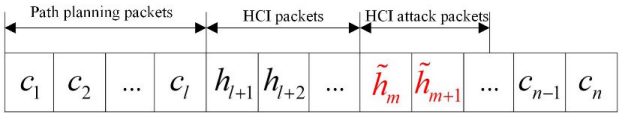


FIGURE 11. The process logic attack based on the HCI device.

motion range of $joint_5$, which is likely to damage its drive. $\tilde{R}$ takes the attacked motion range of joints.

$$\tilde{R} = \begin{cases} -180^\circ \le j_{i_1} \le 180^\circ \\ -35^\circ \le j_{i5} \le 40^\circ \\ 0 \le j_{i3} \le 600mm \\ -35^\circ \le j_{i4} \le 95^\circ \\ -65^\circ \le j_{i2} \le 65^\circ \\ -105^\circ \le j_{i6} \le 25^\circ \end{cases} \tag{21}$$

The case 2 attack model is implemented in the control model. Since a large number of programs in the engineer station use open source code, once the attack programs camouflaged in these code packages are downloaded to the local, it can easily modify the configuration information of the system [39]–[40].

Case 3: At present, the HCI device adopts an autonomous operation mode in a wide range of transport processes and uses HCI mode in a small-space of grab and placement. The two modes are switched at any time by hardwired buttons or PC interface buttons during an operation.

The HCI commands mainly include the position and attitude data of the end joint issued by the operator and the grab or placement commands. In the cosimulation system, we set the key X of the control handle to the HCI device signal input (I >0), set the button Y to the data.mat's data input (I<0). The dial button on the handle triggers the manipulator grab command, and the down button triggers the placement command. It is the same as the software interface. The operation logic of the system is that the position and attitude data, the grab and placement commands are not triggered simultaneously, more specifically, as Equation (22) shown, grab and placement command, only one is valid data in h_i.

$$\tilde{h}_i = (h_{ip}, h_{ia}, h_{ig}, h_{ip}) \tag{22}$$

Figure 11 gives the process logic attack based on the HCI device. For the scenario described in case 3, three process logic attacks are proposed. (1) $\tilde{h}_{ir} = (h_{ip}, h_{ia}, h_{ig}, h_{ip})$, it is the repeated release of position and attitude data between two points. (2) $\tilde{h}_{ip} = (h_{ip}, h_{ia}, h_{ig}, h_{ip})$ is to trigger the placement command before reaching the installation position. (3) $\tilde{h}_{ig} = (h_{ip}, h_{ia}, h_{ig}, h_{ip})$, during the grab process, the placement command is issued.

2) IMPACT ANALYSIS

The impact of the data logic attack mechanism is based on the impact analysis model, which has the ability to show the impact of the attack models on system integrity, security, availability, and accuracy. We present the proposed scenarios in the cosimulation system and get the results illustrated in Figure 12.

Figure 12(a) and Figure 12(b) show the end position and attitude of the cosimulation manipulator without attack.

For the attack $reorder(c_i)$, the control commands with destroyed logic are assigned to the correct manipulator joints. In this attack scenario, the attacked end position and attitude are obtained shown in Figure 12(c) and Figure 12(d). It can

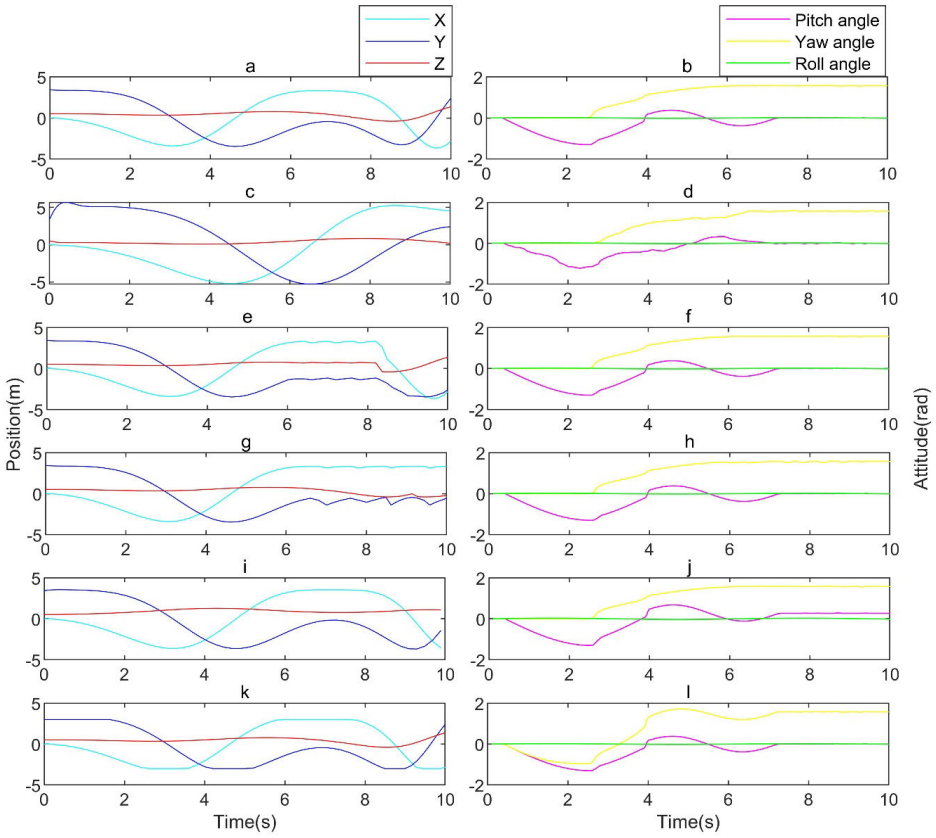


FIGURE 12. The partial results of data logic attacks in the cosimulation.

be seen that the attacked commands make the end position of the manipulator change obviously, and the obtained spatial attitude also changes. In the post-processing function of Adams, the force information of each joint during the motion can be obtained. Combining the information and impact analysis model, we can get $g(\mathrm{I}_i) = 0.52$, $g(\Omega_i) = 0.24$, and $g(\Psi_i) = 2$. This attack breaks the integrity of the transmission data, the security, and the accuracy of the system.

For the attack $replayC_I$, repeatedly sending some of the aforementioned control commands to actuators, the simulation results are shown in Figures 12(e) and 12(f). It can be seen from the figure that the replay attack takes place at 6.2s, changing the manipulator spatial position and attitude. For the attack, there is a significant difference between the desired position and the actual position. Based on the calculation and analysis, we can obtain $g(\mathrm{I}_i) = 0.78$, $g(\Omega_i) = 0.55$, and $g(\Psi_i) = 2$. The $replay(c_i)$ breaks the integrity, security, and accuracy of the transmitted data.

For the attack $drop(c_i)$, data packets are discarded to tamper the data logic. Through Figure 12(g) and the movement of the manipulator during the simulation process, it can be observed that the job is stuck and unable to complete the task. The position error E_{rd} is the ratio of the difference between the actual displacement and the non-attack displacement to the non-attack displacement [41]. In this attack, it is 156%. This attack destroys the system security, the integrity of the transmitted data, and the system accuracy.

For the first attack model provided by case2, we attacked the system configuration matrix B_s in the control model and performed the simulation under the same input signal. The results are shown in Figure 12(i) and Figure 12(j).

Similarly, changing the logic of each joint with the motion ranges can realize a data integrity logic attack. By exchanging two joints' range parameters, the data logic of the HDIM configuration system was destroyed. The simulation results are shown in Figure 12(k) and Figure 12(l). This attack also destroys the system security, the integrity of data transmission, and the system accuracy.

For the process logic attack $\tilde{h}_{ir}$, repeatedly sending the position and attitude data between two points will cause a great impact force. As shown in Figure 13(a), the forward acceleration and reverse acceleration require two opposite forces illustrated in Figure 13(b), which will destroy the security, accuracy, and availability of the system.

As shown in Figure 14, the attack $\tilde{h}_{ip}$ is to trigger the placement commands when the installation position has not arrived, causing the process logic error, which may damage the system and pose a serious threat to the operators and equipment. The attack $\tilde{h}_{ig}$ is to trigger the placement command in the process of grabbing, which will not complete the operation.

TABLE 3. The impact score of data logic attacks.

Scenarios	Attacks	IN		AV			AC		SE		Impact score	Impact ranking
		TR	CI	HA	SA	LA	DI	CI	DA	ET		
Case1	$reorder(c_i)$	1	0	1	1	0	1.68	0.35	0	0	1.01	3
	$replay(c_i)$	1	0	0	1	1	0	0.41	0	0	0.68	6
	$drop(c_i)$	1	0	0	1	1	0	0.52	0	0	0.70	7
Case2	$\tilde{B}_s$	0	1	0	1	1	0	0.68	0	0	0.74	5
	$\tilde{R}$	0	1	0	1	0	0	0.2	0	0	0.44	8
Case3	$\tilde{h}_{ir}$	1	0	0	0	1	0	0.85	1.22	1	1.46	2
	$\tilde{h}_{ip}$	0	0	1	1	0	0	0.73	3	1	2.15	1
	$\tilde{h}_{ig}$	0	0	1	1	0	0	0.96	0	1	0.99	4

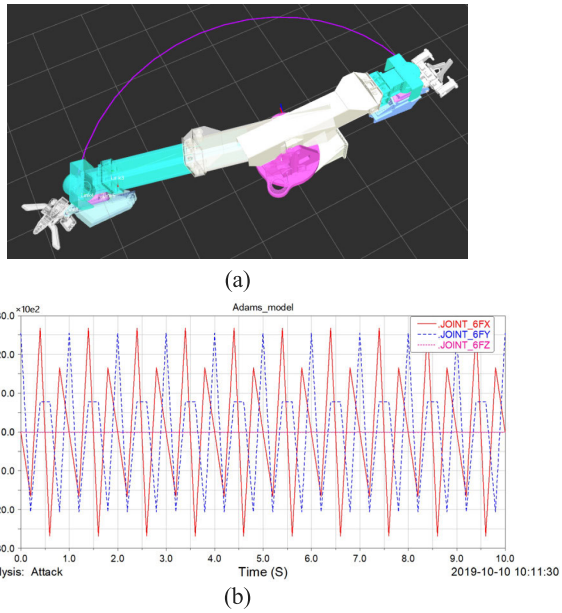


FIGURE 13. The simulation and results. (a) The movement of the manipulator between two points. (b) The forces of end joint under attack.

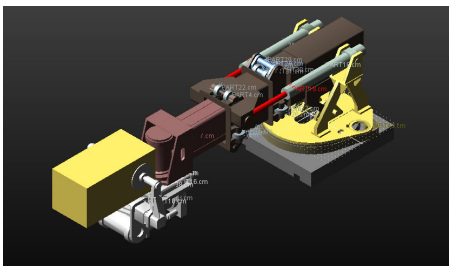


FIGURE 14. The process logic attack experimental environment.

Based on the above analyses and the data obtained from the simulation, Table 3 is obtained to describe the attack impact ranking in the cosimulation. The results show the attack impact and the effectiveness of the data logic attack mechanism in theory. The impact score and impact ranking

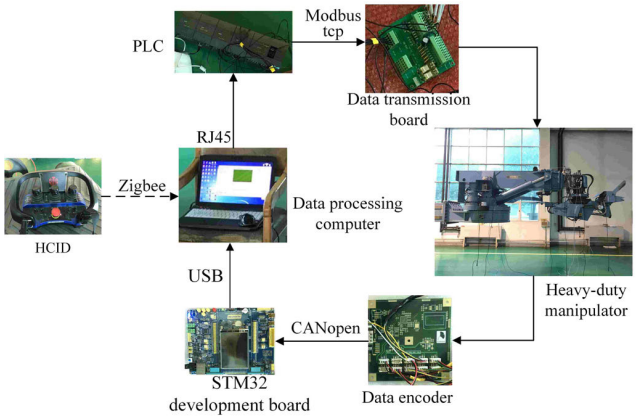


FIGURE 15. The physical test equipment and signal flow interfaces.

FIGURE 16. The process of grab, transport, and placement of the workpiece.

are given in the table. The process logic attack is expected to cause the most significant damage compared to the other two attacks. The network protocol data logic attacks will cause the greatest effect on the AC.

C. PHYSICAL SYSTEM TEST

The physical test system includes an HDIM (working radius 4m, maximum load 3T, freedom 6), a data processing computer, six sike displacement sensors, an STM32 development board, a data transmission board, a PLC, a data encoder, and an HCI device (HCID). Figure 15 shows the physical test equipment and signal flow interfaces.

The HCID communicates with the data processing computer through ZigBee. Then, the data processing computer transmits the HCI commands or motion planning

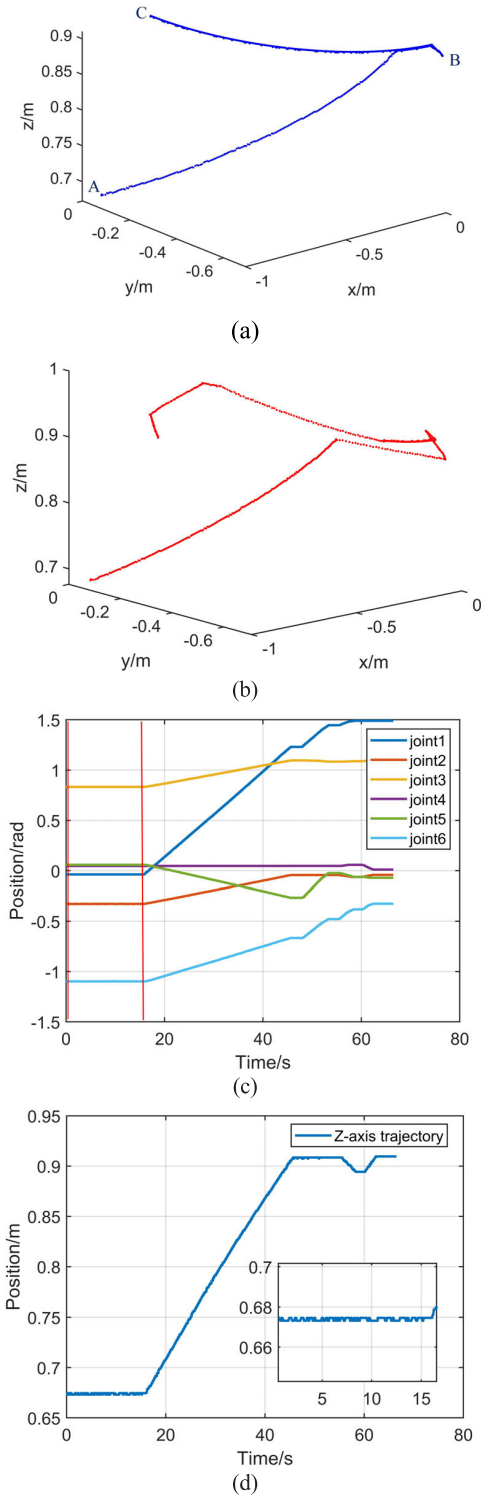


FIGURE 17. The trajectories under attack system data integrity logic. (a) The trajectory of HDIM without attack. (b) The trajectory of HDIM under attack $\tilde{B}_S$. (c) The position of each joint under attack $\tilde{R}$. (d) The trajectory in Z-axis under attack $\tilde{R}$.

commands to the PLC through the network port. Subsequently, the PLC converts the digital data into analog data and transmits the data to the data transmission board via Modbus TCP.

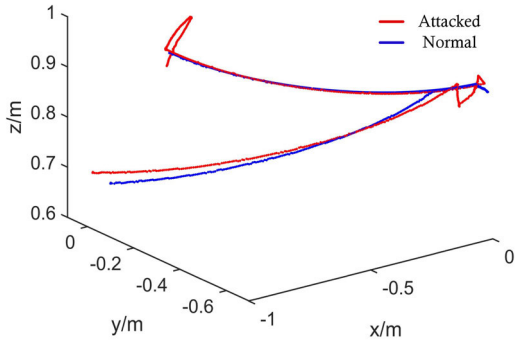


FIGURE 18. The trajectory of HDIM under attack and no attack.

Next, the data have been sent to the HDIM for execution. The data encoder collects the feedback data from sensors and transmits them to the STM32 development board via CANopen. After parsing the CANopen protocol, the development board transmits the parsed data packets to the data processing computer via USB. Finally, the data processing computer performs the next planning motion according to the feedback data.

Figure 16 describes the physical environment of grab, transport, and placement of the workpiece. In order to ensure the security of the large-scale equipment in the physical test system due to its weak security mechanisms, we choose the attacks, whose impact ranking 5, 7, and 8 in Table 3. The corresponding attacks, tampering the logic of system configuration parameters, exchanging the joint movement range of the manipulator, and destroying network protocol logic, have been used to test the impact and attack effectiveness of the proposed data logic attack mechanism based on the cosimulation. In the physical system test, we tested the impact of the above attacks on the HDIM system.

We first deployed the first two attacks under the physical test platform. They belong to system data integrity logic attacks. The results are illustrated in Figure 17.

Figure 17(a) shows the actual space trajectory of the HDIM. The manipulator first grabs the workpiece at position A, transport the workpiece to position B to place it, and goes to the final position c after the placement. (b) shows the end trajectory after changing the system configuration parameters by the attack $\tilde{B}_s$. It is obvious that the operation cannot be completed in the presence of the attack, and there is a jamming phenomenon during the movement. In (c), the movement of each joint under attack at 0-15s. Due to the limitation caused by the exchange of logic parameters $\tilde{R}$, it can be seen that there exists a position limitation phenomenon after the completion of operation planning. It is consistent with Equation (21). (d) illustrates the movement of the manipulator in the Z-axis direction. It can be seen that due to the limitation, the continuous movement of the direction is terminated when the destroyed logic parameters are exceeded. Besides, the burrs in this figure are the joint vibration effect caused by the limitation. Because the HDIM has the characteristics of large inertia, flexible joints, and

rigid connecting rod, the sudden stop of the movement will bring the vibration of flexible joints. In the test, the IN impact score is $g(\mathrm{I}_i) = 1$, the AV impact score is $g(\Phi_i) = 2$, the AC impact score is $g(\Omega_i) = 0.57$, and the SE impact score is $g(\Psi_i) = 0$.

The results of the attack based on the destroyed network protocol logic are shown in Figure 18. The blue line is the normal operation trajectory, and the red line is the trajectory of adding an offset and dropping packets during the motion. Due to the feedback regulation of the HDIM, the target position can still be approached, but the data integrity logic of the network protocol is destroyed, which may cause dangerous situations such as collisions and uncontrollable processes. In this test, the IN impact score is $g(\mathrm{I}_i) = 1$, the AV impact score is $g(\Phi_i) = 2$, the AC impact score is $g(\Omega_i) = 0.25$, and the SE impact score is $g(\Psi_i) = 0$. The test results show the effectiveness of the proposed data logic attack models, which thus verifies the effectiveness of the data logic attack mechanism.

V. CONCLUSION

Based on the analysis of the vulnerabilities of HDIMs, a new data logic attack mechanism is proposed, and the corresponding three data logic attack models are established. Furthermore, this paper proposed an attack impact analysis model to describe the impact and ranking of attacks digitally. Eight attack scenarios about the three data logic attacks were modeled and simulated in the cosimulation, and the impact ranking was given. Subsequently, we established the physical test environment, and the physical test was conducted. The results from the cosimulation and physical test show that the proposed data logic attacks can destroy the security, reliability, accuracy, and security of the HDIM system. Meanwhile, the effectiveness of the data logic attack mechanism has been proven. The data logic attack mechanism and impact analysis model introduced here can be applied to the broader CPS attacks and defense security analyses.

It should be noted that the work is based on the known model, and the data logic attack mechanism based on modeless will bring more concealment. This is what we need to study further.

REFERENCES

[1] L. Kang, S.-H. Kim, W. Kim, and B.-J. Yi, "Review of dimension inhomogeneity in robotics," in *Proc. 16th Int. Conf. Ubiquitous Robots (UR)*, Jun. 2019, pp. 143–148.

[2] J. Lee, P. H. Chang, and M. Jin, "Adaptive integral sliding mode control with time-delay estimation for robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 64, no. 8, pp. 6796–6804, Aug. 2017.

[3] K. Skrzypkowski, W. Korzeniowski, K. Zagórski, and A. Zagórska, "Flexibility and Load-Bearing Capacity of Roof Bolting as Functions of Mounting Depth and Hole Diameter," *Energies*, vol. 12, no. 19, p. 3754, Sep. 2019.

[4] R. Wehbe and R. K. Williams, "Approximate probabilistic security for networked multi-robot systems," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 1997–2003.

[5] T. P. Raptis, A. Passarella, and M. Conti, "Data management in industry 4.0: State of the art and open challenges," *IEEE Access*, vol. 7, pp. 97052–97093, 2019.

[6] M. Pattaranantakul, R. He, Q. Song, Z. Zhang, and A. Meddahi, "NFV security survey: From use case driven threat analysis to state-of-the-art countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3330–3368, 4th Quart., 2018.

[7] C. A. Garcia, D. Lanas, A. M. Edison, S. Altamirano, and M. V. Garcia, "An approach of cyber-physical production systems architecture for robot control," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Washington, DC, USA, Dec. 2018, pp. 2847–2852.

[8] A. Smith. (May 3, 2017). Rogue robots: Testing industrial robot security. Trend Micro. [Online]. Available: https://www.trendmicro.com/vinfo/us/securi-ty/news/internet-of-things/rogue-robots-testing-industrial-robot-security

[9] F. Lorenz and H. Schlingloff, "Online-monitoring autonomous transport robots with an R-valued temporal logic," in *Proc. IEEE 14th Int. Conf. Autom. Sci. Eng. (CASE)*, Munich, Germany, Aug. 2018, pp. 1093–1098.

[10] E. Basan, M. Medvedev, and S. Teterevyatnikov, "Analysis of the impact of denial of service attacks on the group of robots," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Zhengzhou, China, Oct. 2018, pp. 63–71.

[11] H. Fang, L. Xu, Y. Zou, X. Wang, and K.-K.-R. Choo, "Three-stage Stackelberg game for defending against full-duplex active eavesdropping attacks in cooperative communication," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10788–10799, Nov. 2018.

[12] M. Hasan and S. Mohan, "Protecting actuators in safety-critical IoT systems from control spoofing attacks," in *Proc. 2nd Int. ACM Workshop Secur. Privacy Internet-Things (IoT)*, London, U.K., Nov. 2019, pp. 8–14.

[13] T. Huang, D. Zhao, F. Yin, W. Tian, and D. G. Chetwynd, "Kinematic calibration of a 6-DOF hybrid robot by considering multicollinearity in the identification Jacobian," *Mech. Mach. Theory*, vol. 131, pp. 371–384, Jan. 2019.

[14] T. Vuong, A. Filippoupolitis, G. Loukas, and D. Gan, "Physical indicators of cyber attacks against a rescue robot," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PERCOM WORKSHOPS)*, Budapest, Hungary, May 2014, pp. 338–343.

[15] S. Hu, D. Yue, X. Xie, X. Chen, and X. Yin, "Resilient event-triggered controller synthesis of networked control systems under periodic DoS jamming attacks," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4271–4281, Dec. 2019.

[16] X. Sun, R. Nambiar, M. Melhorn, Y. Shoukry, and P. Nuzzo, "DoS-resilient multi-robot temporal logic motion planning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 6051–6057.

[17] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, "Towards the science of security and privacy in machine learning," Nov. 2016, *arXiv:1611.03814*. [Online]. Available: https://arxiv.org/abs/1611.03814

[18] B. Biggio, I. Corona, B. Nelson, B, Rubinstein, and F. Roli, "Security evaluation of support vector machines in adversarial environments," *Support Vector Mach. Appl.*, vol. 12, no. 3, pp. 105–153, Jan. 2014.

[19] K. Ahmad Yousef, A. Almajali, S. Ghalyon, W. Dweik, and B. Mohd, "Analyzing cyber-physical threats on robotic platforms," *Sensors*, vol. 18, no. 5, p. 1643, May 2018.

[20] P. Xun, P.-D. Zhu, Y.-F. Hu, P.-S. Cui, and Y. Zhang, "Command disaggregation attack and mitigation in industrial Internet of Things," *Sensors*, vol. 17, no. 10, p. 2408, Oct. 2017.

[21] Y. Dong, N. Gupta, and N. Chopra, "On content modification attacks in bilateral teleoperation systems," in *Proc. Amer. Control Conf. (ACC)*, Boston, MA, USA, Jul. 2016, pp. 316–321.

[22] G. Sabaliauskaite, G. S. Ng, J. Ruths, and A. Mathur, "Comparison of corrupted sensor data detection methods in detecting stealthy attacks on cyber-physical systems," in *Proc. IEEE 22nd Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Christchurch, New Zealand, Jan. 2017, pp. 235–244.

[23] J. Lin, W. Yu, N. Zhang, X. Yang, and L. Ge, "Data integrity attacks against dynamic route guidance in transportation-based cyber-physical systems: Modeling, analysis, and defense," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8738–8753, Sep. 2018.

[24] T. Bonaci and H. J. Chizeck, "On potential security threats against rescue robotic systems," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, College Station, TX, USA, Nov. 2014, pp. 1–2.

[25] H. Alemzadeh, D. Chen, X. Li, T. Kesavadas, Z. T. Kalbarczyk, and R. K. Iyer, "Targeted attacks on teleoperated surgical robots: Dynamic model-based detection and mitigation," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Toulouse, France, Jun. 2016, pp. 395–406.

[26] F. Maggi, M. Balduzzi, J. Andersson, P. Lin, S. Hilt, A. Urano, and R. Vosseler, "A security evaluation of industrial radio remote controllers," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment.*, Gothenburg, Sweden, Jun. 2019, pp. 133–153.
[27] W. Li, L. Xie, Z. Deng, and Z. Wang, "False sequential logic attack on SCADA system and its physical impact analysis," *Comput. Secur.*, vol. 58, pp. 149–159, May 2016.
[28] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, "An experimental security analysis of an industrial robot controller," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 268–286.
[29] J. P. Monteuuis, J. Petit, J. Zhang, H. Labiod, S. Mafrica, and A. Servel, "Attacker model for connected and automated vehicles," in *Proc. Assoc. Comput. Machinery*, 2018, pp. 1–9.
[30] M. Pogliani, D. Quarta, M. Polino, M. Vittone, F. Maggi, and S. Zanero, "Security of controlled manufacturing systems in the connected factory: The case of industrial robots," *J. Comput. Virol. Hacking Techn.*, vol. 15, no. 3, pp. 161–175, Sep. 2019.
[31] L. Li, L. Xie, W. Li, Z. Liu, and Z. Wang, "Improved deep belief networks (IDBN) dynamic model-based detection and mitigation for targeted attacks on heavy-duty robots," *Appl. Sci.*, vol. 8, no. 5, p. 676, Apr. 2018.
[32] S. Robla-Gomez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *IEEE Access*, vol. 5, pp. 26754–26773, 2017.
[33] P. Corke, "A Simple and systematic approach to assigning Denavit–Hartenberg parameters," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 590–594, Jun. 2007.
[34] T. Handel, M. Schreiber, K. Rothmaler, and G. Ivanova, "Data security and raw data access of contemporary mobile sensor devices," in *Proc. World Congr. Med. Phys. Biomed. Eng.*, Prague, Czech Republic, May 2018, pp. 397–400.
[35] G. Bagyalakshmi, G. Rajkumar, N. Arunkumar, M. Easwaran, K. Narasimhan, V. Elamaran, M. Solarte, I. Hernandez, and G. Ramirez-Gonzalez, "Network vulnerability analysis on brain signal/image databases using Nmap and wireshark tools," *IEEE Access*, vol. 6, pp. 57144–57151, 2018.
[36] C. Cerrudo and L. Apa, "Hacking robots before skynet," in *Proc. IOActive Website*, 2017, pp. 1–17.
[37] L. Li, L. Xie, X. Luo, and Z. Wang, "Compliance control using hydraulic heavy-duty manipulator," *IEEE Trans. Ind. Inf.*, vol. 15, no. 2, pp. 1193–1201, Feb. 2019.
[38] S. Walla and C. Rossow, "MALPITY: Automatic identification and exploitation of tarpit vulnerabilities in malware," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroSP)*, Stockholm, Sweden, Aug. 2019, pp. 590–605.
[39] F. Martín, E. Soriano, and J. M. Cañas, "Quantitative analysis of security in distributed robotic frameworks," *Robot. Auto. Syst.*, vol. 100, pp. 95–107, Feb. 2018.
[40] T. C. H. Kim Kim, J. Rhee, F. Fei, Z. Tu, G. Walkuop, and D. Xu, "RVFUZZER: Finding input validation bugs in robotic vehicles through control-guided testing," in *Proc. 28th USENIX Secur. Symp. (USENIX Secur.)*, San Francisco, CA, USA, Aug. 2019, pp. 425–442.
[41] C. Yu and J. Xi, "Simultaneous and on-line calibration of a robot-based inspecting system," *Robot. Comput.-Integr. Manuf.*, vol. 49, pp. 349–360, Feb. 2018.

LIANPENG LI received the B.S. degree from the Department of Automation, Qingdao University of Technology, in 2014, and the M.S. degree in control engineering from the Beijing University of Information Science and Technology, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Computer and Communication Engineering, University of Science and Technology Beijing, China.

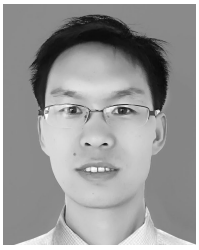
His research interests include cyber-physical systems security and control engineering.

LUN XIE received the M.S. and Ph.D. degrees in control theory and control engineering from the University of Science and Technology, Beijing, China, in 1991 and 2002, respectively.

He is currently a Professor and a Ph.D. Supervisor with the School of Computer and Communication Engineering, University of Science and Technology, Beijing, China. His research interests include cyber-physical systems and active security control of industrial control systems, and theory and application of networked control systems, and artificial intelligence. He is a Board Member of Chinese Association of Artificial Intelligence and the Vice Director of Beijing Society of Internet of Things.

BING HAO received the B.S. degree in mechanical engineering from Chongqing University, in 1988.

He is currently a Senior Manager of CITIC Heavy Industries Company, Ltd. He has completed the development and industrialization of more than ten major equipment and applied them to major national projects. His research interests include machinery manufacturing and automation, and industrial robot systems.



LIUSONG YANG received the B.S. and M.S. degrees in industrial robot system development materials science and engineering from the Henan University of Science and Technology, in 2004 and 2007, respectively.

He is currently a Senior Engineer of CITIC Heavy Industries Company, Ltd. His research interests include industrial robot system design and machine design.

TONGHAI HU received the B.S. degree in mechanical engineering from Jilin University, in 1988.

For more than 30 years, he has been engaged in the design of heavy-duty manipulators. He is currently a Technical Expert of CITIC Heavy Industries Company, Ltd. The manipulator described in this paper is designed by him. His research interests include machine design, hydraulic control, and mechanical simulation.

ZHILIANG WANG received the M.S. and Ph.D. degrees in control theory and control engineering from Yanshan University, Harbin Institute of Technology, in 1982 and 1989, respectively.

He is currently a Professor and Ph.D. Supervisor with the School of Computer and Communication Engineering, University of Science and Technology, Beijing, China. His research interests include security control of cyber physical systems and active industrial control systems, and theory and application of networked control systems, and artificial intelligence. He is a Senior Board Member of Chinese Association of Artificial Intelligence and the Director of Beijing Society of Internet of Things.