IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 29, 2019, accepted October 21, 2019, date of publication October 25, 2019, date of current version November 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949645

An Enhanced Multi-Stage Semantic Attack Against Industrial Control Systems

YAN HU[1], YUYAN SUN[2,3], YOUCHENG WANG[4], AND ZHILIANG WANG[1]

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2]Beijing Key Laboratory of IoT Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China
[3]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100195, China
[4]Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, Beijing Electro-Mechanical Engineering Institute, Beijing 100074, China

Corresponding author: Yan Hu (huyan@ustb.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61802016 and Grant 61702506, in part by the National Key Research and Development Program of China under Grant 2017YFB0802805, in part by the National Social Science Foundation of China under Grant 17ZDA331, in part by the Project funded by China Postdoctoral Science Foundation under Grant 2018M641198, and in part by the Fundamental Research Funds for the Central Universities under Grant FRF-BD-18-016A.

ABSTRACT Industrial Control Systems (ICS) play a very important role in national critical infrastructures. However, the growing interaction between the modern ICS and the Internet has made ICS more vulnerable to cyber attacks. In order to protect ICS from malicious attacks, intrusion detection technology emerges. By analyzing the network meta data or the industrial process data, Intrusion Detection Systems (IDS) can identify attacks that violate communication protocols or system specifications. However, the existing intrusion detection technology is not omnipotent, which opens up a back door for some more advanced attacks. In this work, we design an enhanced multi-stage semantic attack against ICS, which is undetectable by existing IDS. By hijacking the communication channels between the Human Machine Interface (HMI) and the remote Programmable Logic Controllers (PLCs), the attacker can manipulate the measurement data and control instructions simultaneously. The fake measurement data deceives the human operator into making wrong decisions. Furthermore, the attacker can strategically manipulate the semantic meaning of control instructions according to system state transition rules. In the meanwhile, a fake view of measurement data is presented to the HMI to conceal the on-going malicious attack. This attack is totally stealthy since the message sizes and timing, the command sequences, and the system state values are all legitimate. Consequently, this attack can secretly bring the system into critical states. Experimental results have verified the strong attack ability of the proposed attack.

INDEX TERMS Industrial control systems, multi-stage semantic attacks, state transition, stealthy attacks.

I. INTRODUCTION

Nowadays, Industrial control systems (ICS) [1] play a quite important role in a variety of industrial processes, such as manufacturing, public facilities (e.g., buildings and airports), power generation and distribution [2]–[4], chemical processing [5], water treatment [6], oil and gas transportation [7], or large-scale communication [8]. The rapid development of Internet Technology (IT) facilitates ICS to realize remote process control and intelligent decision making. However, high exposure to open networks has made ICS an attractive target for malicious attackers [9], [10]. The summer of 2010 was a landmark to ICS security. By that time the core control program of the Natanz uranium enrichment base in Iran was infected by an unprecedented sophisticated cyber worm called ''Stuxnet''. The centrifuge for uranium enrichment was forced to accelerate unconventionally and was eventually damaged, which caused a huge loss to the entire nuclear plant. In 2015, the notorious Trojan malware ''BlackEnergy3'' attacked the Ukrainian power grid. False commands sent to relays triggered unconventional circuit disconnections, immediately followed by a large-scale blackout. At Black Hat 2017 [11], Dr. Staggs pointed out that cyber and physical attacks can invade the programmable automation controllers and OPC (OLE for Process Control) servers easily by exploiting the wind farm design and implementation flaws. Additionally, they designed corresponding attack tools to launch attacks on actual wind farms. So many ICS security

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

incidents indicate that ICS security has become a critical global issue [12], [13].

Intrusion Detection Systems (IDS) provide a promising solution for protecting ICS [14], [15]. IDS are a type of software designed to find indications that information systems have been compromised. Traditional intrusion detection technology is mainly classified into two categories, signature-based and anomaly-based. Signature-based IDS, also called misuse-based, build a blacklist containing the signatures of known attacks, and raise alarms when the system behavior matches any of these signatures. Anomaly-based IDS are mainly used to detect anomalies that violate the normal behavior patterns of a target system. Therefore, a normal behavior model of the target system should be constructed. Model parameters can be learnt from unaffected system operating data. While applying intrusion detection to ICS, the industrial process data (e.g., measurement data and control instructions) is another important factor to consider [16]. If the value of a process variable is outside its normal range or breaks the fundamental laws of nature, an alarm should be raised.

Exiting intrusion detection technology is proved to be useful but not omnipotent. Recently, Kleinmann *et al.* [17] have proposed a multi-stage semantic attack against ICS. This attacker can drive the target system to a critical state by reversing the semantic meaning of control instructions and presenting a fake view of measurement data to the system operator at the same time. However, the attacker cannot guarantee to realize the attack goal, since it just randomly chooses some instructions to reverse. In this work, we design an enhanced and strategic multi-stage semantic attack against ICS, which relies on the system state transition rules to precisely decide which control instructions to reverse. The enhanced semantic attack can significantly improve the attack success rate while maintaining its stealthiness.

The key contributions of this work are summarized as follows:

- We analyze the relationships between system states and control instructions, and build a system state transition graph that can accurately characterize the dynamic behavior of ICS.
- We design an enhanced multi-stage semantic attack against ICS. By exploiting system state transition rules, the attacker can develop accurate attack strategies, which can increase the attack success rate significantly.
- We launch the enhanced multi-stage semantic attack on a simulated industrial control system to verify its stronger attack ability compared to the existing semantic attack.

The rest of the paper is organized as follows. We introduce the research literature about intrusion detection in Section II. Some preliminaries of the enhanced semantic attack are presented in Section III. In Section IV, we elaborate on the principles of the enhanced multi-stage semantic attack against ICS. Experiments are conducted in Section V to verify the stronger attack ability of the enhanced multi-stage semantic attack. Finally, a conclusion is drawn in Section VI.

II. RELATED WORK

Due to the growing openness of ICS, cyber attacks against traditional information systems also threaten the security of ICS. Traditional intrusion detection technology mainly fall into two classes: signature-based and anomaly-based. The former mainly relies on the accurate signatures of malicious attacks. System behavior that matches any existing attack signature is considered anomalous. On the contrary, the latter depends on a normal behavior model. Any system behavior that deviates from this model should be flagged as an anomaly. Generally speaking, attacks against ICS usually violate protocol specifications or cause abnormal network traffics, and the physical constraints of ICS are likely to be broken during attack. Therefore, we introduce the intrusion detection technology on ICS from three aspects: network protocol analysis, network traffic mining, and process data analysis.

A. NETWORK PROTOCOL ANALYSIS-BASED INTRUSION DETECTION

Network protocols define a set of rules to specify how network devices should format, transmit and process information. Therefore, intrusion detection rules can be extracted from network protocols. Any system behavior that violates the detection rules is judged to be abnormal. Some open protocols are commonly used in ICS communication, e.g., ModBus, DNP3, ICCP/TASE.2. These protocols are vulnerable to a variety of malicious attacks such as eavesdropping, tampering and counterfeiting, since ICS were designed to run in relatively closed environments and security was rarely considered in the design of industrial communication protocols.

Cheung *et al.* [18] extracts a normal system behavior model from the industrial protocol specifications. The model formalizes legal data values and legal relationships between different data fields. Furthermore, a set of communication modes are built according to data transmission ports, transmission directions and security requirements of ICS. Any behavior that violate the normal behavior model or the communication modes should be flagged as an anomaly, so this detection technique also belongs to the anomaly-based intrusion detection. Morris *et al.* [19] construct signatures for ModBus protocol vulnerabilities by exploiting a famous intrusion detection system—*Snort*. Communication data that matches any of these signatures is identified as an anomaly. Moreover, traditional IDS can be tailored or improved for intrusion detection on ICS. Lin *et al.* [20] successfully realize intrusion detection on ICS by implanting a DNP3 protocol parser into *Bro*, a network intrusion detection system developed by the University of Berkeley.

In addition to open protocols, proprietary protocols also play an important part in ICS communication. IDS based on proprietary protocol analysis has emerged. Hong *et al.* [21]

extract specifications from the IEC 61850 standards (e.g., Generic Object Oriented Substation Event (GOOSE) and Sample Value technology (SV)), based on which to identify abnormal or malicious behaviors in electric power substations. In [22], legal and illegal network traffic patterns are defined based on the protocol specifications of power systems. These patterns are further converted into Snort rules for intrusion detection.

As described above, intrusion detection based on network protocol analysis mainly relies on the accurate definition of detection rules, and usually yields a high false alarm rate and incurs a large message-parsing time overhead. Intrusion detection based on network traffic mining can overcome these shortcomings to some extent.

B. NETWORK TRAFFIC MINING-BASED INTRUSION DETECTION

Most ICS have fixed business logics, static and simple network topologies, and a small number of programs. Therefore, traffics in industrial networks are stable in most cases. Unusual traffic patterns generally indicate the occurrence of an anomaly, which is the main motivation of the network traffic mining-based intrusion detection.

Traditional IDS based on network traffic mining [23] mainly rely on the analysis of network meta data, including IP addresses (i.e., source IP address for outbound packets and destination IP address for inbound packets), transmission ports, traffic durations, and packet intervals. Applying data mining techniques to network meta data can identify system anomalies effectively. Supervised [24] and semi-supervised [25] clustering, single-class [26] or multi-class [27] support vector machine, mixed Gaussian model [28], fuzzy logic [29]–[31], neural network [32], [33] and deep learning [34] are commonly used techniques for traffic mining. These techniques aim to model the non-linear relationships between network traffics and system behaviors. The relationship model and real-time traffic data are used to investigate the current status of the system, and then detect malicious attacks timely. However, analyzing a large number of traffic features undoubtedly incurs a high computational overhead. Therefore, techniques like principal component analysis [35] and ant colony optimization [36] are used to remove redundant traffic features, thus to reduce computational overhead.

Intrusion detection techniques based on protocol analysis and traffic mining are borrowed from the traditional network intrusion detection domain. They are mainly designed for conventional information systems. A big difference between ICS and the traditional information systems (i.e., ICS are closely related to the physical world) makes these techniques difficult to identify attacks against physical processes, since these attacks may not violate network protocol specifications or cause abnormal network traffics. Hence, the intrusion detection technology based on process data analysis has emerged.

C. PROCESS DATA ANALYSIS-BASED INTRUSION DETECTION

Industrial process data is another important information source for intrusion detection on ICS. It is likely for a system operator to make wrong decisions [37] if the process data is secretly counterfeited or tampered with, and eventually cause lethal damage to ICS. Generally, the deviation between the observed and expected process values can determine whether an attack has occurred [38]. In [39], all process variables are divided into three classes: constants, enumeration, and continuous values. Each process variable has a normal behavior pattern. Once the monitored value of a process variable does not conform to its normal behavior pattern, an alarm is raised. In [40], system states are denoted by measurement data reported by a group of remote sensors, and a corresponding state distance measurement method is presented. Anomalies can be detected by inspecting the distance between the current state and the critical states.

Time series forecasting provides another potential solution for intrusion detection on ICS. This technology can precisely predict the future outputs of ICS, which are then compared with the monitored outputs to generate residuals. By applying proper statistical techniques to the residuals, IDS can detect malicious attacks effectively. In general, the residual series conforms to a Gaussian distribution during normal operation of ICS. If an attack occurs, there will be a significant deviation between the actual and expected system behaviors, i.e., the residuals deviate from 0 notably [41]. Two kinds of intrusion detection techniques based on residual analysis are summarized in [42]: sequential detection and change detection. The first technique can identify anomalies as quickly as possible. In other words, it determines the shortest residual sequence based on which IDS can make a judgement. The second technique identifies an anomaly if the residual [43] or the cumulative residual [16] exceeds a predefined threshold at a certain time point.

Recently, Kleinmann *et al.* [17] propose a multi-stage semantic attack against ICS by tampering with the measurement data and the control instructions simultaneously. They state that the Modbus protocol has no security protection mechanism or message integrity protection mechanism, which opens up a back door for malicious attackers. This vulnerability enables the adversary to reverse the semantic meaning of control instructions and present a fake view of measurement data to the HMI at the same time. However, this attack is sometimes futile, because it cannot exactly decide which control instructions to manipulate. Randomly reversing some instructions cannot guarantee to realize the attack goal. In this work, we design an enhanced multi-stage semantic attacks against ICS, which makes full use of the system state transition rules and strategically decides which control instructions to reverse, thus to bring the target system into dangerous situations precisely. The enhanced semantic attack is totally undetectable by traditional IDS because all process values are legal during this attack. Additionally, it can improve the attack success rate significantly when compared

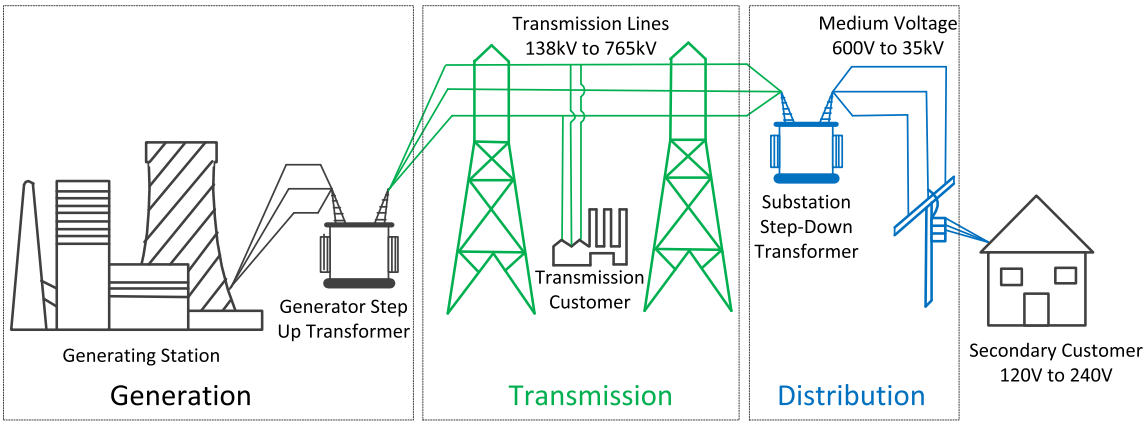


FIGURE 1. The Electricity Distribution Subsystem (Following [17]).

to the existing instruction-reversing semantic attack proposed in [17].

III. PRELIMINARIES

In this section, we present some preliminaries of the enhanced semantic attack, including the communication mechanism of Modbus, the architecture of the electrical distribution system—a typical industrial system, and the underlying adversary model.

A. MODBUS

ModBus is a de facto application layer protocol for ICS. This protocol supports a master-slave communication mode between different control devices, even if they are within different types of buses or networks. Most Modbus systems use TCP as the communication layer protocol. A Modbus/TCP message is embedded in TCP segments and TCP port 502 is reserved for Modbus communications. In Modbus communications, usually the HMI acts as the unique master and the remote PLCs act as slaves. In a transaction, the master requests process data from the slaves or issues control instructions to the slaves. The slaves respond by sending the requested data to the master or by performing the control instructions. The request message from the master contains a unique transaction ID, which should be contained in the corresponding response message.

A Modbus Protocol Data Unit (PDU) consists of two fields: a single-byte Function code and a variable-size Payload (limited to 252 bytes). The Function code specifies the operation to be taken, and the Payload contains parameters required by the function invocation. For example, the Payload of a read request consists of two fields, a reference number and a bit/word count. The former specifies the starting memory address for reading. The latter specifies the number of memory object units to be read. The payload of the corresponding response message is comprised of two parts: byte count and data, which respectively record the length of data in bytes and the data contents that were read. In addition to the starting memory address, the payload of a write message has another field that specifies the data to be written.

Unfortunately, Modbus has little ability to defend itself against malicious attacks, e.g., data tampering or counterfeiting. Moveover, Modbus only uses TCP sequence numbers to provide simple session semantics, but cannot ensure message integrity or long-term session semantics. Therefore, TCP session hijacking becomes quite straightforward.

B. ELECTRICITY DISTRIBUTION SYSTEM

An electricity supply chain is typically comprised of three subsystems: generation, transmission, and distribution, as illustrated in Fig. 1. The transmission network connects the generation system with the distribution system. Electricity is transmitted from generation sites to remote distribution substations along high-voltage transmission lines. The high voltage (138 kV to 765 kV) is then converted to medium-voltage (600V to 35kV) by substation transformers. A group of medium-voltage circuits fan out from the substation. The medium voltage is further stepped down to the low voltage (commonly 120/240V) by the distribution transformers close to end users. In this work, we mainly discuss the distribution subsystem between the substations and distribution transformers, which is the target system of the ‘‘BlackEnergy" cyber-attack.

In order to improve reliability, distribution circuits are usually equipped with ‘‘tie switches’’ (also called switchgears, which are normally disconnected) to other circuits. If one of the circuits encounter an unintentional fault, it will be connected to another circuit by an adjacent switchgear. Thus, electricity flows into the faulted circuit and some necessary services are restored. The switchgears can be operated automatically or manually from the HMI. A simplified model of the subsystem is shown in Fig. 2. Two medium-voltage circuits fan out from the substation. There are six PLCs (i.e., PLC01 $\sim$ PLC06) along the top circuit and four PLCs (i.e., PLC08 $\sim$ PLC11) along the bottom circuit. Additionally, the two distribution lines are interconnected by a normally open switchgear that is controlled by PLC07.

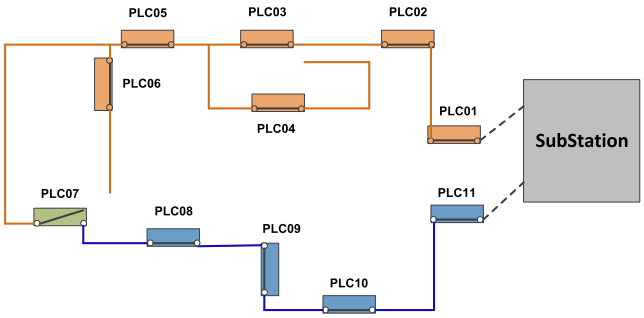


FIGURE 2. The Electricity Distribution Subsystem.

C. ADVERSARY MODEL

In the adversary model, we suppose that the attacker can penetrate into the control network and launch a Man-In-The-Middle (MITM) attack between the HMI and remote PLCs. On the hijacked communication link, all network packets can be eavesdropped, replayed, delayed or deleted before reaching their destinations. Furthermore, the attacker can modify the packet contents and even take over the HMI to fabricate malicious control instructions. The goal of the adversary is to disrupt the normal operation of ICS and cause fatal damages to the physical system.

Furthermore, suppose that the adversary has gained sufficient knowledge of the ICS architecture, the industrial process and the way to manipulate the target system. Here, we use a somewhat weaker type of attack model: the attacker can penetrate into the control network, and launch MITM attacks on one or more HMI-PLC communication links simultaneously. However, this model is assumed to be stateless, i.e., it does not tamper with TCP sequence numbers. Therefore, this model cannot delete existing messages or inject fake ones. It can only manipulate the contents of existing packets.

IV. ENHANCED MULTI-STAGE SEMANTIC ATTACK

In this section, we elaborate on the strategy of the enhanced multi-stage semantic attack against ICS.

A. DEFINITION OF SYSTEM STATES

Suppose that an electricity distribution subsystem involves a set of configurable state variables that is denoted by $\{x_1, x_2, \ldots, x_N\}$, where N is the total number of state variables, and $x_i \in \{1, -1\}$ $(1 \leq i \leq N)$ is the ith state variable, which denotes the status (closed or open) of the ith switchgear. Hence, a state vector $\boldsymbol{x}$ can be used to represent the status of the entire system at a certain time point:

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_N), \tag{1}$$

All possible values of the state vector $\boldsymbol{x}$ constitute a set $\mathcal{X}$. In the electricity distribution subsystem, $\mathcal{X}$ is comprised of three mutually exclusive subsets, a normal state set $\mathcal{N}$, a fault state set $\mathcal{F}$ and a critical state set $\mathcal{C}$. The normal states in $\mathcal{N}$ indicate that the system is operating normally. If there occur some unavoidable disturbance or system faults, the system enters a fault state contained in $\mathcal{F}$ to restore some necessary services and finally return to the normal state. However, if the system encounters some malicious attacks, it will be brought into some dangerous or unwanted situations (i.e., critical states), like large-scale blackouts.

The normal state set $\mathcal{N}$ of the electricity distribution system is formalized as follows:

$$\mathcal{N} = \{\boldsymbol{x}^{Nor_1}, \boldsymbol{x}^{Nor_2}, \ldots, \boldsymbol{x}^{Nor_L}\}, \tag{2}$$

where $\mathcal{N} \subset \mathcal{X}$, L is the total number of the normal state vectors, and $\boldsymbol{x}^{Nor_l}$ $(1 \leq l \leq L)$ is the lth normal state vector, which consists of the values of N state variables:

$$\boldsymbol{x}^{Nor_l} = (x_1^{Nor_l}, x_2^{Nor_l}, \ldots, x_N^{Nor_l}). \tag{3}$$

Analogously, the fault state set and critical state set are defined by:

$$\mathcal{F} = \{\boldsymbol{x}^{Fau_1}, \boldsymbol{x}^{Fau_2}, \ldots, \boldsymbol{x}^{Fau_K}\}, \tag{4}$$

and

$$\mathcal{C} = \{\boldsymbol{x}^{Cri_1}, \boldsymbol{x}^{Cri_2}, \ldots, \boldsymbol{x}^{Cri_M}\}, \tag{5}$$

where $\mathcal{F}$ and $\mathcal{C}$ are two subsets of $\mathcal{X}$ (i.e., $\mathcal{F} \subset \mathcal{X}$, $\mathcal{C} \subset \mathcal{X}$), K and M are the numbers of fault states and critical states, respectively. Furthermore, the fault state vector and the critical state vector are defined by:

$$\boldsymbol{x}^{Fau_k} = (x_1^{Fau_k}, x_2^{Fau_k}, \ldots, x_N^{Fau_k}), \tag{6}$$

and

$$\boldsymbol{x}^{Cri_m} = (x_1^{Cri_m}, x_2^{Cri_m}, \ldots, x_N^{Cri_m}). \tag{7}$$

where $x_i^{Fau_k}$ $(1 \leq k \leq K$ and $1 \leq i \leq N)$ denotes the ith entry of the kth fault state vector, and $x_j^{Cri_m}$ $(1 \leq m \leq M$ and $1 \leq j \leq N)$ denotes the jth entry of the mth critical state vector. The three subsets $\mathcal{N}$, $\mathcal{F}$ and $\mathcal{C}$ are mutually exclusive and together constitute the entire state set $\mathcal{X}$, i.e., $\mathcal{N} \cap \mathcal{F} = \mathcal{N} \cap \mathcal{C} = \mathcal{F} \cap \mathcal{C} = \varnothing$ and $\mathcal{N} \cup \mathcal{F} \cup \mathcal{C} = \mathcal{X}$.

B. SYSTEM STATE TRANSITION

Based on the definition of system states, we now define the state transition rules. Suppose that the system operator can configure the target system manually, i.e., issue the "open" or "close" instructions to change the status of switchgears. Therefore, we use a variable $a \in \{-1, 1, 0\}$ to denote different operations the system operator can take on a switchgear. The values -1, 1, and 0 represent the "open", "close" and no action, respectively. Suppose that there are N operable switchgears in the system, corresponding to N configurable state variables mentioned above. A N-tuple vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_N)$ is used to represent all operations taken by the system operator at a certain time point. Each entry $a_i \in \{-1, 1, 0\}$ denotes the operation taken on the ith state variable x_i.

State transition rules describe how the system behavior changes over time. We use $x_i(t)$ and $x_i(t+1)$ to denote

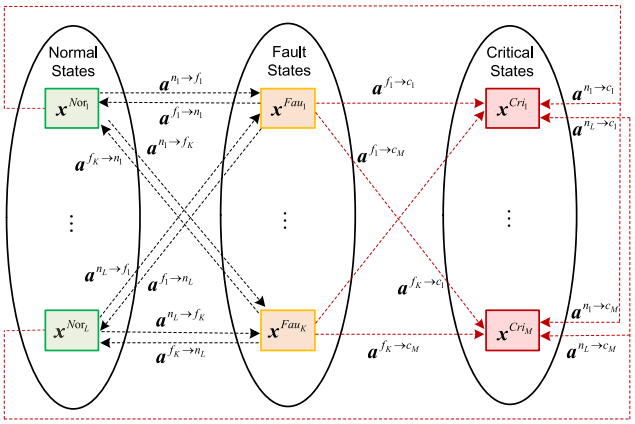


FIGURE 3. The System State Transition Graph.

the current state and the next state of the ith switchgear, respectively. An operation $a_i(t)$ can drive $x_i(t)$ to $x_i(t+1)$, so we formalize the state transition of a swithgear as follows:

$$x_i(t+1) = x_i(t) \otimes a_i(t), \tag{8}$$

where the operator $\otimes$ defines the following rule:

$$x_i(t+1) = \begin{cases} a_i(t), & \text{if } a_i(t) \neq 0, \\ x_i(t), & \text{otherwise.} \end{cases} \tag{9}$$

This equation indicates that the next state $x_i(t+1)$ is determined jointly by the current state $x_i(t)$ and the current operation $a_i(t)$. If no operation is taken (i.e., $a_i(t) = 0$), $x_i(t+1)$ is set equal to $x_i(t)$. Otherwise, $x_i(t+1)$ is set equal to $a_i(t)$. Therefore, the state transition of the entire system can be formalized by:

$$\boldsymbol{x}(t+1) = \boldsymbol{x}(t) \otimes \boldsymbol{a}(t), \tag{10}$$

where the state transition of each element of the state vector $\boldsymbol{x}$ follows Eq. 8. The state transition graph is illustrated in Fig. 3. A normal state transits to a fault state if some unavoidable disturbances or faults occur. A fault state can return to a normal state after the necessary services are restored. However, if the target system encounters a malicious attack, it is likely to enter a critical state from a normal state or a fault state.

C. ATTACK STRATEGY

With the definition of system states and state transition rules, we now describe the strategy of the enhanced multi-stage semantic attack against ICS. The attack strategy mainly consists of measurement data deception and control instruction manipulation. During measurement data deception, a fake view of process data is presented to the HMI, thus to induce the system operator to take some unnecessary operations. Afterwards, the issued instructions are tampered with by the attacker to achieve specific attack goals. Below we elaborate on the two attack steps.

1) MEASUREMENT DATA DECEPTION

During measurement data deception, the attacker can change the measurement data, e.g., current and voltage values reported by victim PLCs, to any legitimate value, thus to bypass IDS. Suppose that the victim PLCs are those controlling the top line in Fig. 2 (i.e., PLC01 to PLC06). The left graph in Fig. 4 shows the actual values of the current and voltage reported by PLC01. The right graph depicts the fake values of the same measurement data presented to the HMI. When the system is attacked (from 240s to 270s), zero current and zero voltage are presented to the HMI. The fake view simulates a natural fault on the top line, so it is not regarded as a malicious attack. In other words, the attack is totally stealthy. The fake view misleads the system operator into taking uncessary remediation measures, which may be costly and harmful. Furthermore, it provides the attacker a good opportunity to manipulate the control instructions maliciously.

2) CONTROL INSTRUCTION MANIPULATION

Once the system operator observes the zero current and zero voltage reported by remote PLCs for a period of time, he will drive the system to a fault state by issuing specific control instructions. Suppose that a set of control instructions that is denoted by $\boldsymbol{a}^{n_l \to f_k}$ is issued to change the status of one or more switchgears. At this moment, the attacker can change the vector $\boldsymbol{a}^{n_l \to f_k}$ to a malicious one $\boldsymbol{a}^{n_l \to c_m}$ before the instructions reach their destinations, thus bringing the system into a critical state. Here, $\boldsymbol{a}^{n_l \to f_k}$ and $\boldsymbol{a}^{n_l \to c_m}$ are the operation vectors that can drive the system from the normal state to a fault state and a critical state, respectively. In order to bypass intrusion detection, the tampered instructions should meet the following two conditions: 1) $|\boldsymbol{a}^{n_l \to f_k}| = |\boldsymbol{a}^{n_l \to c_m}|$ and 2) $\boldsymbol{a}^{n_l \to f_k} \neq \boldsymbol{a}^{n_l \to c_m}$, where $|\boldsymbol{a}| = (|a_k|)_{1 \leq k \leq N}$ denotes the vector of absolute values of $\boldsymbol{a}$'s elements. Thus, no existing instruction is dropped and no fabricated instruction is injected. Additionally, all instruction values remain legitimate in the tampered messages, so the attack is totally stealthy.

If the attacker fails to manipulate the instructions in this step, he has another chance. When the system has restored the necessary services, it should return to the normal state from the fault state once the system operator issues the corresponding instructions $\boldsymbol{a}^{f_k \to n_l}$. At this moment, the attacker can rewrite $\boldsymbol{a}^{f_k \to n_l}$ into a malicious vector $\boldsymbol{a}^{f_k \to c_m}$, in order to bring the system into a critical state. Analogously, $\boldsymbol{a}^{f_k \to c_m}$ should satisfy $|\boldsymbol{a}^{f_k \to n_l}| = |\boldsymbol{a}^{f_k \to c_m}|$ and $\boldsymbol{a}^{f_k \to n_l} \neq \boldsymbol{a}^{f_k \to c_m}$. Once the system enters a critical state, the attack goal is achieved.

The entire procedure of the Enhanced Multi-Stage Semantic Attack (EM2SA for short) is summarized in Algorithm 1. The normal, fault and critical system state sets are used as inputs to the algorithm. The output of the algorithm is a boolean variable *flag* that indicates whether the semantic attack is successful or not. The initial value of *flag* is set to *false*, as shown in line 1. Lines 2 and 3 make

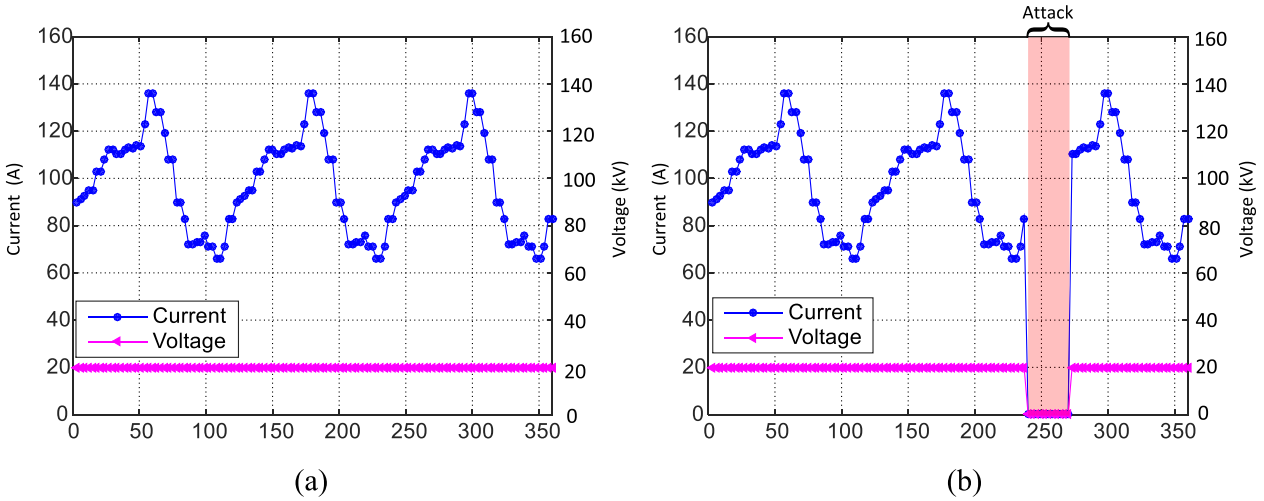


FIGURE 4. Measurement Data Deception Attack.

Algorithm 1 *EM2SA* Algorithm

Input: The normal, fault and critical system state collections $\mathcal{N}$, $\mathcal{F}$ and $\mathcal{C}$
Output: a *flag* indicating whether the attack is successful or not

1 $flag \leftarrow false$;
2 construct the state transition graph $\mathcal{G}$;
3 Penetrate the control network to get a Man-In-The-Middle position;
4 launch the measurement data deception attack when $state_{system} \in \mathcal{N}$;
5 **while** *true* **do**
6 tamper with the control instruction $\boldsymbol{a}^{n_l \to f_k}$ to $\boldsymbol{a}^{n_l \to c_m}$, that satisfies $|\boldsymbol{a}^{n_l \to f_k}| = |\boldsymbol{a}^{n_l \to c_m}|$ and $\boldsymbol{a}^{n_l \to f_k} \neq \boldsymbol{a}^{n_l \to c_m}$;
7 wait for the system state transition;
8 **if** $state_{system} \in \mathcal{C}$ **then**
9 $flag \leftarrow true$;
10 break;
11 **else**
12 launch the measurement data deception attack;
13 tamper with the new control instruction $\boldsymbol{a}^{f_k \to n_l}$ to $\boldsymbol{a}^{f_k \to c_m}$, that satisfies $|\boldsymbol{a}^{f_k \to n_l}| = |\boldsymbol{a}^{f_k \to c_m}|$ and $\boldsymbol{a}^{f_k \to n_l} \neq \boldsymbol{a}^{f_k \to c_m}$;
14 wait for the system state transition;
15 **if** $state_{system} \in \mathcal{C}$ **then**
16 $flag \leftarrow true$;
17 break;
18 **end**
19 **end**
20 **end**
21 return *flag*;

some preparations, including building the state transition graph and getting a Man-In-The-Middle position in the control network. Lines 4 to 20 are the whole procedure of the semantic attack. Line 4 launches the measurement data deception attack when the system operates normally, which presents a fake view of the measurement data to the HMI. Afterwards, the attacker tampers with the instructions issued by the system operator and waits for the system state transition (lines 6 and 7). If this attack is successful (i.e., the system enters a critical state: $state_{system} \in \mathcal{C}$), the output variable *flag* is set to *true* and the attack procedure ends (lines 8 to 10). Otherwise, the attacker has another chance to manipulate the control instructions when the system is going back to the normal state, as shown in lines 11 to 19. If both the two attacks are unsuccessful, the attacking procedure should be restarted, and line 20 returns the output variable *flag*.

V. EXPERIMENTS AND DISCUSSION

In this section, we simulate the above-mentioned electricity distribution subsystem in Java language and launch two different semantic attacks on the simulated system. The architecture of the simulated ICS is depicted in Fig. 2, including a substation and two radial distribution lines, each with a group of PLCs. One virtual machine is used to simulate the HMI, which acts as the Modbus master. Other virtual machines simulate the remote PLCs, which serve as the Modbus slaves. On the simulated system, we launch two attacks: the enhanced multi-stage semantic attack proposed in this work and the instruction-reversing semantic attack proposed in [17], and compare the success rate of the two attacks.

We present the normal current values reported by three key PLCs (PLC01, PLC07 and PLC11) and the normal voltage value reported by PLC01 in Fig. 5. The voltage value remains stable, while the current values measured by PLC01 and PLC11 vary with the changing loads. The switchgear controlled by PLC07 keeps open when the system operates normally, so the current reported by PLC07 is zero.

Fig. 6a and Fig. 6b respectively show the fake measurement data presented to the HMI and the actual measurement data when the system encounters the instruction-reversing

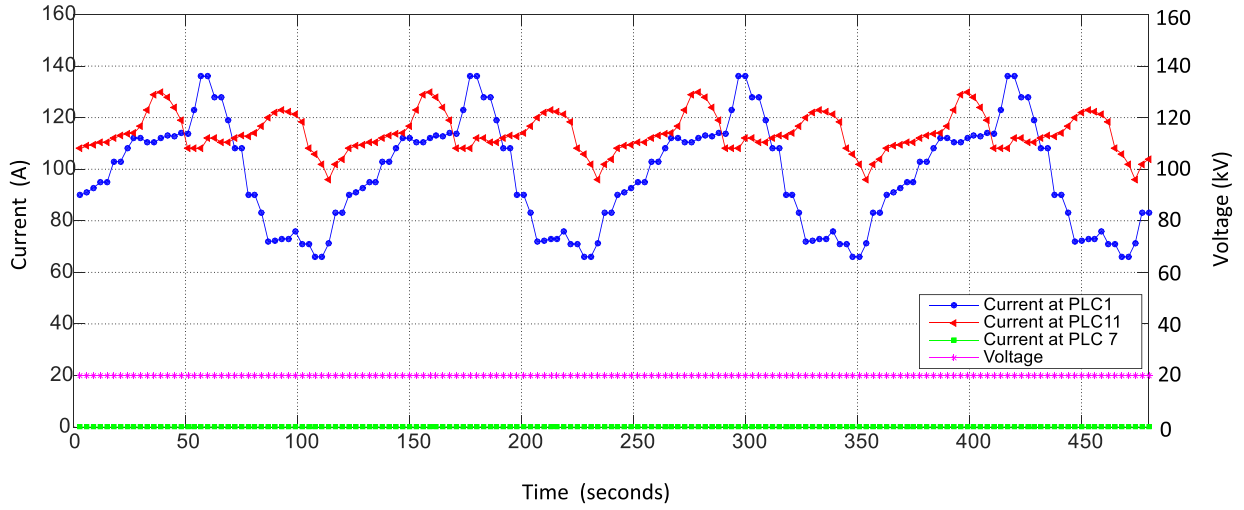


FIGURE 5. Normal Measurement Data.

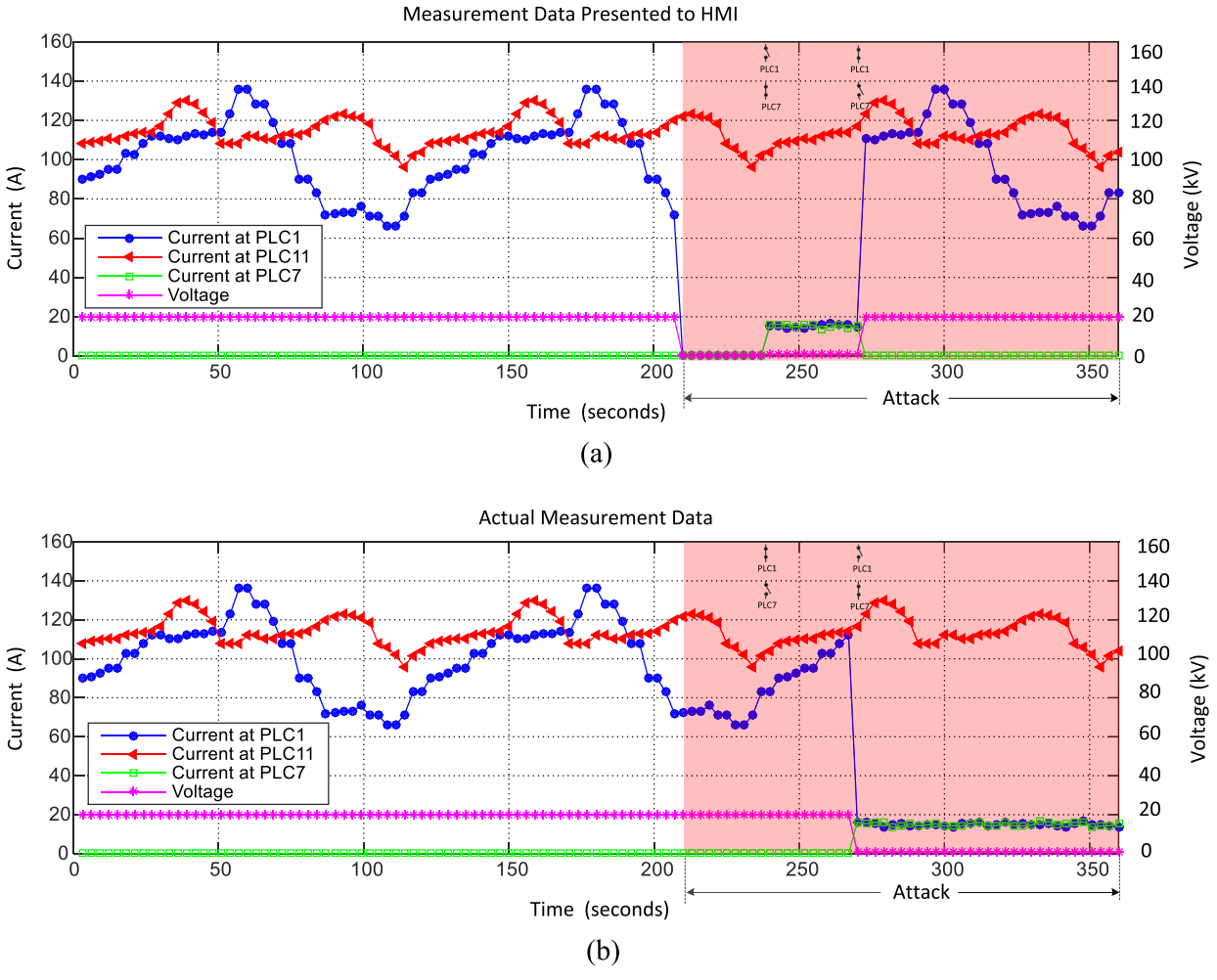


FIGURE 6. The Fake and Actual Measurement Data during the Instruction-Reversing Semantic Attack [17].

semantic attack proposed in [17]. As we can see from Fig. 6a, the measurement data deception starts at 210s. After that, the system operator observes the zero current and zero voltage at PLC01 on the HMI. Therefore, the system operator issues control instructions to open the switchgear controlled by PLC01 and close the switchgear controlled by PLC07 at 240s.

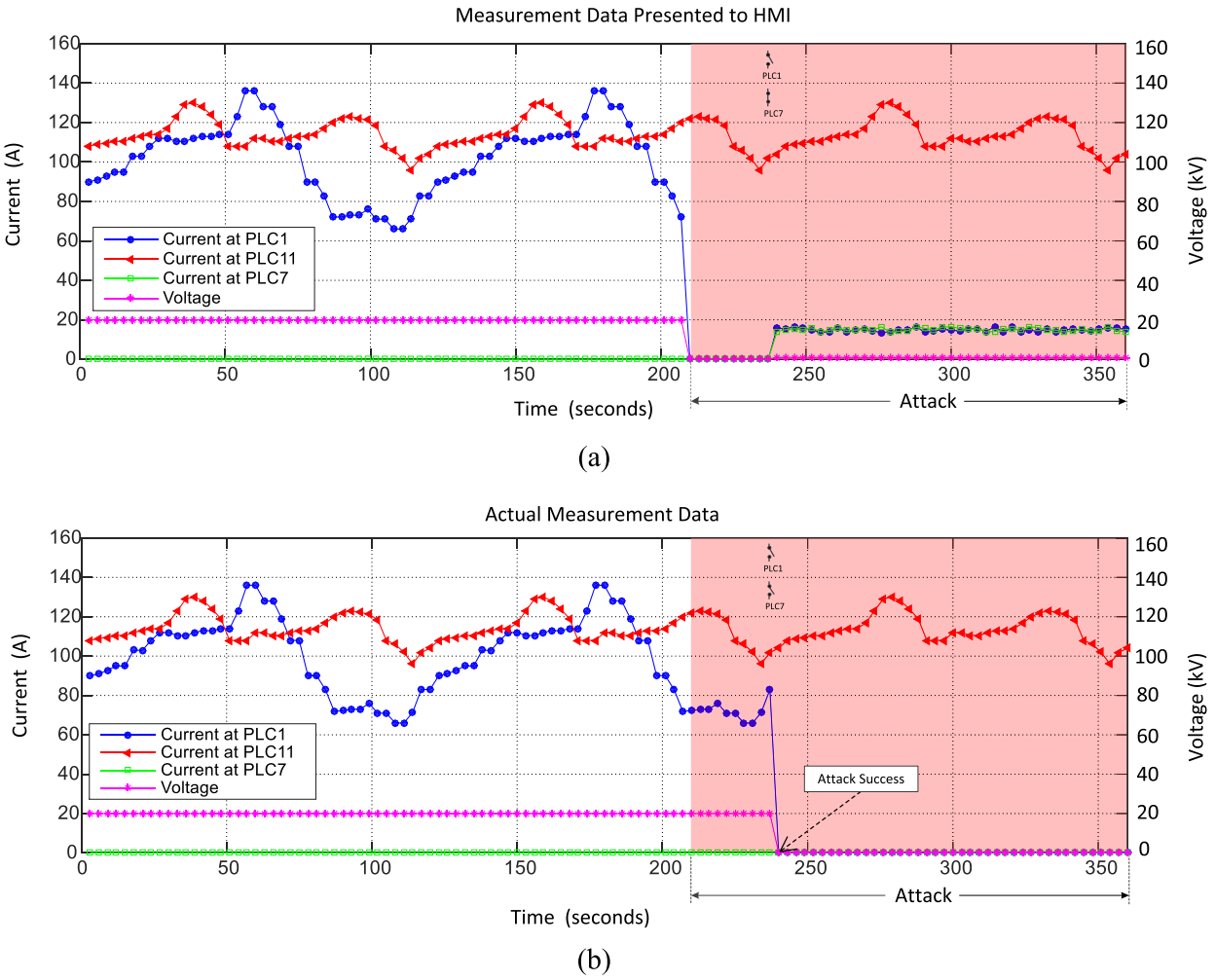


FIGURE 7. **The Fake and Actual Measurement Data during the Enhanced Semantic Attack that Succeeds by One-Step Instruction Tampering.**

Thus the system enters a fault state and the top line begins to restore necessary services. After 240s, the HMI is still provided with a fake view of the measurement data: small values of the current and voltage at PLC01, misleading the system operator into believing the system is being restored. After a period of time, the operator issues control instructions to connect the switchgear controlled by PLC01 and disconnect the switchgear controlled by PLC07 at 270s, in order to bring the system back to normal. Afterwards, the attacker shows the normal current and voltage values to the HMI, presenting an illusion that the system has returned to normal. However, the actual status of the system is shown in Fig. 6b. The attacker reverses each control instruction at 240s and 270s. In detail, the switchgears controlled by PLC01 and PLC07 are respectively closed and opened at 240s, and then respectively opened and closed at 270s. Therefore, the two switchgears maintain the status quo from 240s to 270s, and the measurement data are normal during this period. From 270s, the system enters a superfluous fault recovery phase, so the currents at PLC01 and PLC07 and the voltage at PLC01 are significantly smaller than their normal values. Therefore, the attack goal is not achieved since the system does not enter a critical state.

Fig. 7 shows the fake and actual measurement data during the enhanced multi-stage semantic attack proposed in this work. Firstly, we suppose that the first-step instruction tampering succeeds. Similar to Fig. 6a, Fig. 7a shows that the measurement data deception starts at 210s. After tampering with the "fault recovery" instructions successfully, the attacker presents the small current and voltage values to the HMI after 240s, misleading the system operator into believing the system is being restored. However, as shown in Fig. 7b, the attacker manipulates the instructions strategically at 240s according to Algorithm 1, i.e., reversing the instruction sent to PLC01 while keeping the instruction sent to PLC07 unchanged, in order to bring the system into a critical state. Hence, the actual current and voltage at PLC01 become zero at 240s, which indicates a blackout on the top transmission line, so the attack goal is achieved.

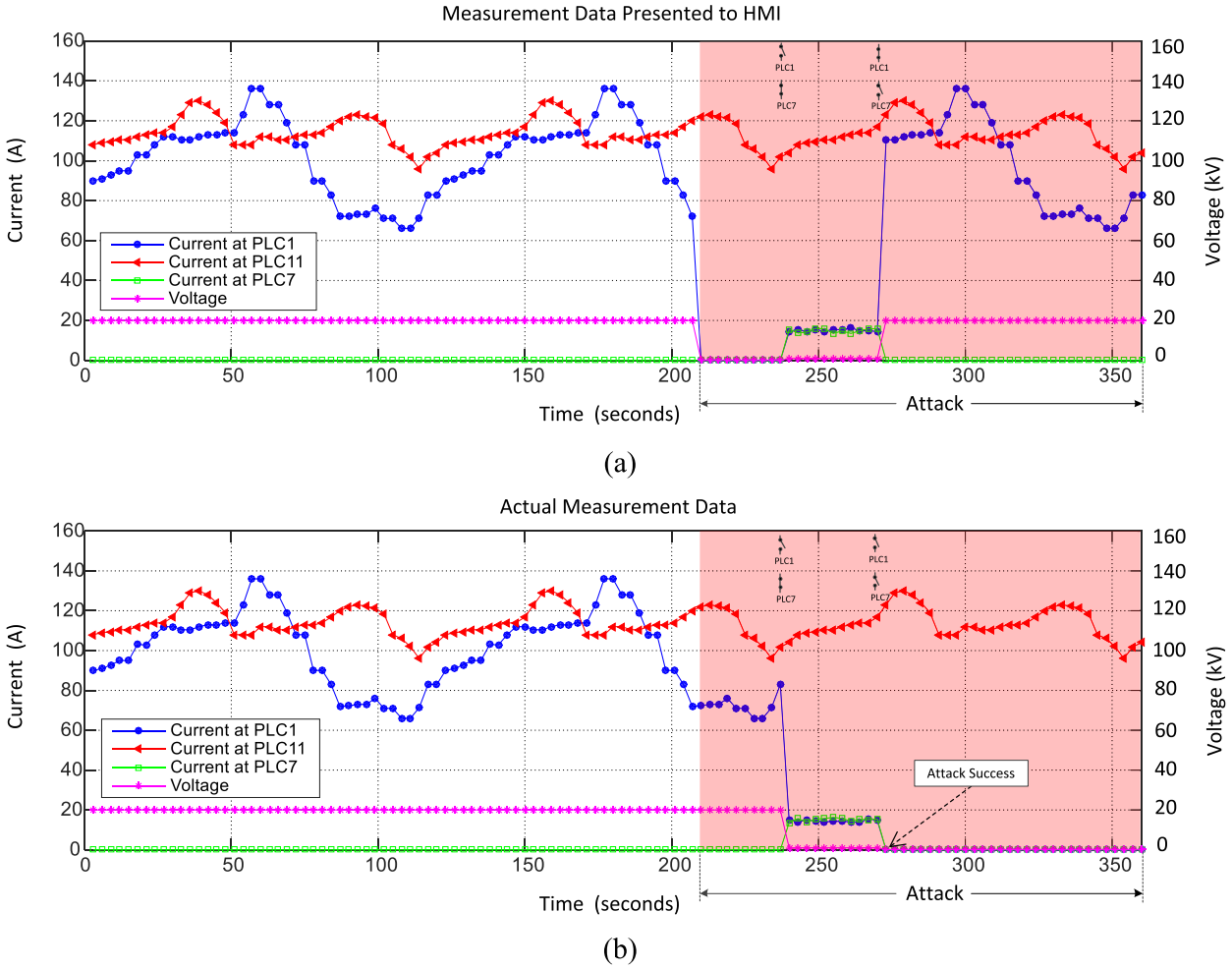


FIGURE 8. The Fake and Actual Measurement Data during the Enhanced Semantic Attack that Succeeds by Two-Step Instruction Tampering.

If the first-step instruction tampering is unsuccessful, the attacker has another chance. As depicted in Fig. 8, the attacker fails to tamper with the control instructions at 240s, but succeeds to manipulate the instruction sent to PLC01 at 270s. Therefore, the system enters a critical state after 270s (both the current and voltage at PLC01 become zero), as shown in Fig. 8b, but the fake measurement data presented to the HMI are normal after 270s, as shown in Fig. 8a. Figs. 7 and 8 indicate that there are two possible paths from the normal state to a critical state during the enhanced multi-stage semantic attack, which are represented by the two red dashed lines in Fig. 9.

Specially, if the attacker can randomly choose one or more instructions to tamper with during the instruction-reversing semantic attack proposed in [17], the proposed enhanced semantic attack is a special case of that kind of attack. Additionally, suppose that each instruction tampering attack has a Possibility of Failure (PoF for short). Based on the assumptions, we compare the success rate of the two kinds of semantic attacks on the simulated system. The instruction-reversing semantic attack can randomly choose whether to reverse

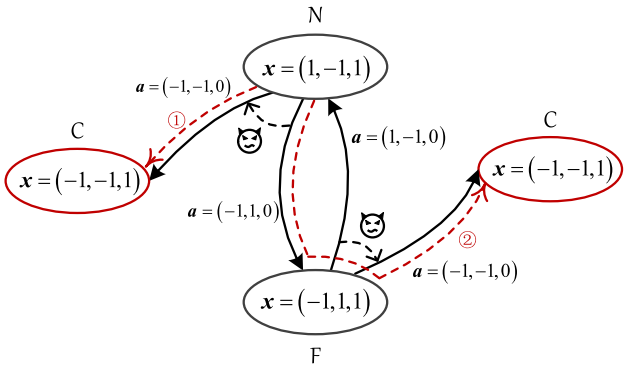


FIGURE 9. Two Attack Paths During the Enhanced Semantic Attack.

an eavesdropping instruction, while the enhanced semantic attack manipulates an instruction strategically according to Algorithm 1. In this experiment, PoF varies from 0.1 to 0.9, with a step value of 0.1. For each value of PoF, we conduct 5000 simulations for each attack. The comparison of the two attacks is illustrated in Fig. 10. Obviously, the success rate of the enhanced multi-stage semantic attack is significantly

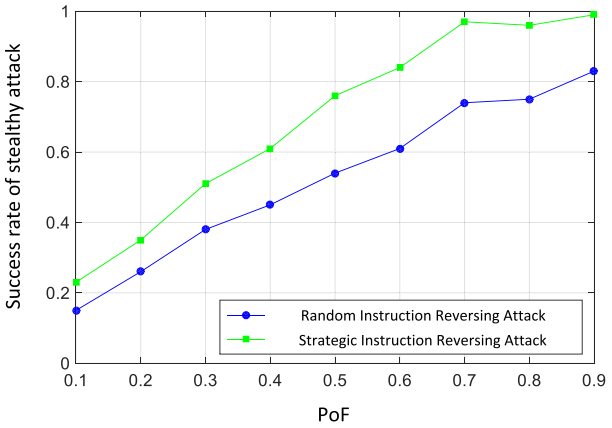


FIGURE 10. **Comparison of Attack Success Rates of Two kinds of Attacks.**

higher than that of the instruction-reversing attack, which verifies the stronger attack ability of the enhanced attack.

VI. CONCLUSION

In this paper, we propose an enhanced multi-stage semantic attack against ICS. During this attack, a fake view of measurement data is first presented to the HMI to mislead the system operator into issuing unnecessary control instructions. Thus, the attacker has chances to manipulate the control instructions strategically according to system state transition rules, and precisely bring the target system into a critical state. In the meanwhile, the measurement data deception attack should be continued in order to conceal the on-going attack. Furthermore, this attack is totally stealthy, since the command sequences, message sizes, and process values all remain legitimate. To verify the strong attack ability of the enhanced multi-stage semantic attack, we simulate an electricity distribution subsystem in Java language. Additionally, we compare the attack success rate of the enhanced semantic attack with that of the existing instruction-reversing semantic attack. The experimental results show that the enhanced semantic attack can significantly improve the attack success rate. In future research, we will try to investigate the proposed attack on some real-world and large-scale ICS testbeds and seek for effective countermeasures against this kind of attacks, e.g., securing the communication channel via cryptographic means, e.g., by adding data integrity protections such as digital signatures or message authentications to prevent the attacker from modifying packets.

REFERENCES

[1] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ics) security," *NIST Special Publication*, vol. 800, no. 82, p. 16, 2011.

[2] J. Tian, R. Tan, X. Guan, and T. Liu, "Enhanced hidden moving target defense in smart grids," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2208–2223, Mar. 2019.

[3] Y. Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proc. IEEE*, vol. 100, no. 1, pp. 195–209, Jan. 2012.

[4] T. Liu, Y. Liu, Y. Mao, Y. Sun, X. Guan, W. Gong, and S. Xiao, "A dynamic secret-based encryption scheme for smart grid wireless communication," *IEEE Trans. Smart Grid*, vol. 5, no. 3, pp. 1175–1182, May 2014.

[5] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *J. Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[6] J. Weiss, "Industrial control system (ics) cyber security for water and wastewater systems," in *Securing Water and Wastewater Systems*. Cham, Switzerland: Springer, 2014, pp. 87–105.

[7] M. R. Akhondi, A. Talevski, S. Carlsen, and S. Petersen, "Applications of wireless sensor networks in the oil, gas and resources industries," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Apr. 2010, pp. 941–948.

[8] A. D. Papadopoulos, A. Tanzman, R. A. Baker, Jr., R. G. Belliardi, and D. J. Dube, "System for remotely accessing an industrial control system over a commercial communications network," U.S. Patent 6 061 603 A, May 9, 2000.

[9] R. K. Koehler, "When the lights go out: Vulnerabilities to us critical infrastructure, the russian cyber threat, and a new way forward," *Georgetown Secur. Stud. Rev*, vol. 7, no. 1, pp. 27–36, 2018.

[10] L. Maglaras, K. H. Kim, H. Janicke, M. A. Ferrag, S. Rallis, P. Fragkou, A. Maglaras, and T. J. Cruz, "Cyber security of critical infrastructures," *ICT Exp.*, vol. 4, no. 1, pp. 42–45, 2018.

[11] J. Staggs, *Adventures in Attacking Wind Farm Control Networks*. Las Vegas, NV, USA: Black Hat, 2017. [Online]. Available: https://www.blackhat.com/docs/us-17/wednesday/us-17-Staggs-Adventures-In-Attacking-Wind-Farm-Control-Networks.pdf

[12] D. Ding, Q.-L. Han, Y. Xiang, C. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, Jan. 2018.

[13] Z. Ling, K. Liu, Y. Xu, Y. Jin, and X. Fu, "An end-to-end view of IoT security and privacy," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.

[14] P. Haller and B. Genge, "Using sensitivity analysis and cross-association for the design of intrusion detection systems in industrial cyber-physical systems," *IEEE Access*, vol. 5, pp. 9336–9347, 2017.

[15] Z. Zhang, H. Zhu, S. Luo, Y. Xin, and X. Liu, "Intrusion detection based on state context and hierarchical trust in wireless sensor networks," *IEEE Access*, vol. 5, pp. 12088–12102, 2017.

[16] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the impact of stealthy attacks on industrial control systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 1092–1105.

[17] A. Kleinmann, O. Amichay, A. Wool, D. Tenenbaum, O. Bar, and L. Lev, "Stealthy deception attacks against SCADA systems," in *Computer Security*. Berlin, Germany: Springer, 2017, pp. 93–109.

[18] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proc. SCADA Secur. Sci. Symp.*, vol. 46, 2007, pp. 1–12.

[19] T. Morris, R. Vaughn, and Y. Dandass, "A retrofit network intrusion detection system for modbus rtu and ascii industrial control systems," in *Proc. 45th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2012, pp. 2338–2345.

[20] H. Lin, A. Slagell, C. Di Martino, Z. Kalbarczyk, and R. K. Iyer, "Adapting bro into SCADA: Building a specification-based intrusion detection system for the dnp3 protocol," in *Proc. 8th Annu. Cyber Secur. Inf. Intell. Res. Workshop*, Jan. 2013, p. 5.

[21] J. Hong, C.-C. Liu, and M. Govindarasu, "Detection of cyber intrusions using network-based multicast messages for substation automation," in *Proc. ISGT*, Feb. 2014, pp. 1–5.

[22] H. Hadeli, R. Schierholz, M. Braendle, and C. Tuduce, "Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2009, pp. 1–8.

[23] P. Stavroulakis and M. Stamp, *The Handbook of Communication and Security*. Cham, Switzerland: Springer, 2010.

[24] C.-H. Tsang and S. Kwong, "Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Dec. 2005, pp. 51–56.

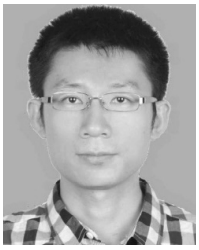
[25] H. Wang, "On anomaly detection and defense resource allocation of industrial control networks," Ph.D. dissertation, College Control Sci. Eng., Zhejiang Univ., Hangzhou, China, 2014.

[26] L. A. Maglaras and J. Jiang, "Intrusion detection in SCADA systems using machine learning techniques," in *Proc. Sci. Inf. Conf. (SAI)*, Aug. 2014, pp. 626–631.

[27] Y. Luo, "Research and design on intrusion detection methods for industrial control system," Ph.D. dissertation, College Control Sci. Eng., Zhejiang Univ., Hangzhou, China, 2013.
[28] I. Kiss, B. Genge, and P. Haller, "A clustering-based approach to detect cyber attacks in process control systems," in *Proc. 13th IEEE Int. Conf. Ind. Inform. (INDIN)*, Jul. 2015, pp. 142–148.
[29] O. Linda, M. Manic, T. Vollmer, and J. Wright, "Fuzzy logic based anomaly detection for embedded network security cyber sensor," in *Proc. IEEE Symp. Comput. Intell. Cyber Secur. (CICS)*, Apr. 2011, pp. 202–209.
[30] O. Linda, M. Manic, J. Alves-Foss, and T. Vollmer, "Towards resilient critical infrastructures: Application of Type-2 Fuzzy Logic in embedded network security cyber sensor," in *Proc. 4th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2011, pp. 26–32.
[31] O. Linda, M. Manic, and T. Vollmer, "Improving cyber-security of smart grid systems via anomaly detection and linguistic domain knowledge," in *Proc. 5th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2012, pp. 48–54.
[32] T. Vollmer and M. Manic, "Computationally efficient neural network intrusion security awareness," in *Proc. 2nd Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2009, pp. 25–30.
[33] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2009, pp. 1827–1834.
[34] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. 9th EAI Int. Conf. Bio-Inspired Inf. Commun. Technol. (Formerly BIONETICS) (ICST)*, May 2016, pp. 21–26.
[35] C. Hou, J. Hanhong, W. Rui, and L. Liu, "A probabilistic principal component analysis approach for detecting traffic anomaly in industrial networks," *J. Xi'an Jiaotong Univ.*, vol. 46, no. 2, pp. 78–83, 2012.
[36] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 420–432, May 2016.
[37] M. Krotofil, J. Larsen, and D. Gollmann, "The process matters: Ensuring data veracity in cyber-physical systems," in *Proc. 10th ACM Symp. Inf., Comput. Commun. Secur.*, Apr. 2015, pp. 133–144.
[38] E. Colbert, D. Sullivan, S. Hutchinson, K. Renard, and S. Smith, "A process-oriented intrusion detection method for industrial control systems," in *Proc. 11th Int. Conf. Cyber Warfare Secur.* New York, NY, USA, Academic, 2016, p. 497.
[39] D. Hadžiosmanović, R. Sommer, E. Zambon, and P. H. Hartel, "Through the eye of the PLC: Semantic security monitoring for industrial processes," in *Proc. 30th Annu. Comput. Secur. Appl. Conf.*, Dec. 2014, pp. 126–135.
[40] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta, "A multidimensional critical state analysis for detecting intrusions in SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 179–186, May 2011.
[41] R. J. Patton, "Robustness in model-based fault diagnosis: The 1995 Situation," *Annu. Rev. Control*, vol. 21, pp. 103–123, Jan. 1997.
[42] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," in *Proc. 6th ACM Symp. Inf., Comput. Commun. Secur.*, Mar. 2011, pp. 355–366.
[43] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 580–591, Mar. 2014.

YAN HU was born in 1988. She received the B.S. degree in automation from Xi'an Jiaotong University, Xi'an, Shannxi, China, in 2011, and the Ph.D. degree in computer science from the University of Chinese Academy of Sciences, Beijing, China, in 2017.

Since 2017, she has been an Assistant Professor with the University of Science and Technology Beijing, China. Her main research interests include security of industrial control systems, security of the Internet of Things, and service computing.



YUYAN SUN was born in 1982. He received the B.S. degree in computer science from Peking University, Beijing, China, in 2004, and the M.S. and Ph.D. degrees in computer science from University of Chinese Academy of Sciences, Beijing, in 2007 and 2016, respectively.

From 2007 to 2012, he was a Research Assistant with the Institute of Software, Chinese Academy of Sciences. Since 2016, he has been an Assistant Professor with the Beijing Key Laboratory of IoT Information Security, Institute of Information Engineering, Chinese Academy of Sciences, and School of Cyber Security, University of Chinese Academy of Sciences. His main research interests include security of industrial control systems and the IoT.

YOUCHENG WANG received the B.S. degree in telecommunications engineering from Hubei University, Wuhan, China, in 2011, and the Ph.D. degree in electromagnetic wave and microwave technology from the University of Chinese Academy of Sciences, Beijing, China, in 2016.

From 2011 to 2016, he was a member of the Laboratory of Electromagnetic Radiation and Sensing Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing. He is currently with the Beijing Electro-Mechanical Engineering Institute, Beijing. His research interests include the Internet of Things, ultrawideband (UWB) antennas and array antenna, electromagnetic scattering characteristics, and the application of UWB radar.

ZHILIANG WANG was born in 1956. He received the B.S. degree in industrial automation from Yanshan University, Qinhuangdao, Hebei, China, in 1982, and the M.S. and Ph.D. degrees in electronic engineering from the Harbin Institute of Technology, Harbin, Heilongjiang, China, in 1985 and 1988, respectively.

From 1988 to 1991, he was a Postdoctoral Researcher with Zhejiang University. Since 1991, he has been a Professor and a Ph.D. Supervisor with the University of Science and Technology Beijing, China. His main research interests include the Internet of Things and robot technology.