electronics

MDPI

Article

False Sequential Command Attack of Large-scale Cyber-Physical Systems

Yinqiao Xiong [1,2], Ziyu Yang [3,*], Baoyao Wang [1], Peng Xun [1] and Tiantian Deng [1,2]

1 College of Computer, National University of Defense Technology, Changsha 410073, China; yq.xiong@ccsu.edu.cn (Y.X.); wangbaoyao16@nudt.edu.cn (B.W.); xunpeng12@nudt.edu.cn (P.X.); dtt@ccsu.edu.cn (T.D.)

2 Department of Electronic Information and Electrical Engineering, Changsha University, Changsha 410022, China

3 Institute of systems engineering, Academy of Military Sciences, Beijing 100141, China

* Correspondence: ziyuyang_academy@outlook.com; Tel.: +86-133-9760-8869

Received: 31 July 2018; Accepted: 31 August 2018; Published: 4 September 2018

Abstract: Previous studies have demonstrated that false commands can cause severe damage to large-scale cyber-physical systems (CPSs). We focus on a kind of threat called false sequential command attack, with which attackers can generate false sequential commands, resulting in the illegal control of the physical process. We present a feasible attack model. Attackers delay the disaggregation of former commands by manipulating maliciously sub-controllers. Simultaneously, bad feedback data is injected to defeat the controller to issue latter commands. Thus, false command sequence is executed and the disruption of physical process can be obtained. It is also difficult for the detector to identify such attacks as injecting bad data. We also discuss other possible attack paths and analyze the corresponding disadvantages. Compared with other paths, the proposed model is more feasible and has more difficulties to be detected. A case study is given to validate the feasibility and effectiveness of proposed false sequential command attack model. Finally, we discuss the possible countermeasure.

Keywords: cyber-physical system; sequential command; security; threat model; false data injection

1. Introduction

A cyber-physical system (CPS) is the tight combination of physical system and information system, including a controller, actuators, sensors, critical state estimation, detector, and communication system [1,2]. A simplified CPS model is described in Figure 1. The controller issues command to control actuators according to the estimated state. Actuators operate in the physical world and the physical process is sensed by sensors. Sensors transmit the sensory data to the critical state estimation, and then the critical state estimation evaluates the current state of the physical system. The detector collects commands from the controller and the sensory data to judge whether the system state or command is legal. Once an illegal state or command occurs, an alarm is given.

However, with the wide open of communication infrastructure which is used to improve efficiency, reliability, and sustainability of supply [3] such as smart grid, the new vulnerability has been exposed [2]. The attackers can utilize the vulnerability to destroy CPS, and the power outages in Ukrainian is one of the examples [4]. There mainly exist three attack entry points as shown in Figure 1: attacking sensors such as false data injection [5–8], attacking controllers such as false command injection [9], and attacking communication system such as jamming the communication channel (DOS attack) [10–12] and time-delay attack [13,14].

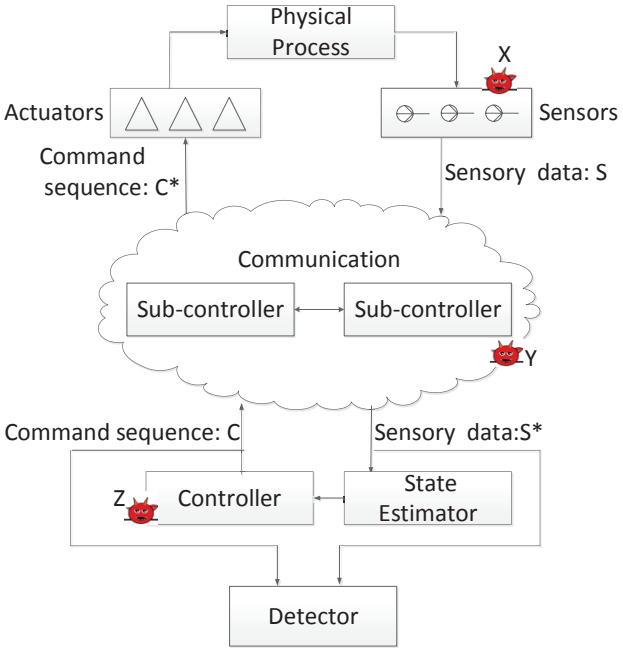


Figure 1. A simple model of CPS, where *X*, *Y* and *Z* denote attack entry points [15]. Adapted with permission from [15], Copyright Elsevier, 2016.

In this paper, we mainly focus on one threat called false sequential command attack. False sequential commands refer to a set of commands whose legal order is disordered leading to the false control. For example, in Figure 2, under the normal situation, the command that turns on the pump needs to be executed after turning on the valve. This kind of attack tries to delay the time of turning on the valve until the command turning on the pump is executed, which may cause the disruption of physical components and even have a blast.

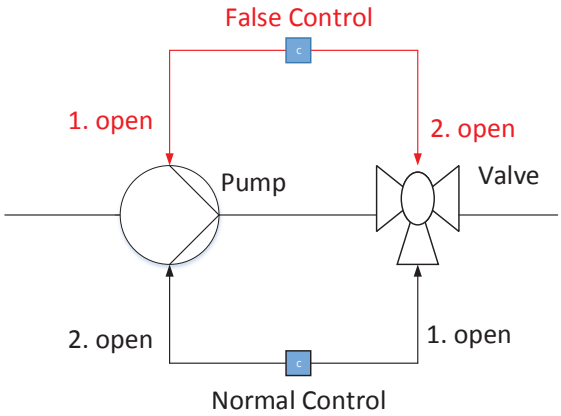


Figure 2. A part of a chemical reactor system [15]. Adapted with permission from [15], Copyright Elsevier, 2016.

Previous work [15] has introduced the false sequential command attack, which demonstrates the situation that the order of command sequence can be modified and a false command sequence can lead to the disruption of the physical process. However, the effective attack path is not described. This kind of attack may be difficult to get effective impact because controllers with critical state estimation are often intelligent to issue the legal command sequence, which means the order of two successive

commands may be difficult to be altered. For example, in Figure 2, after turning on valve command occurs, the information that the valve has been opened is sent to the controller. If the controller does not receive the information, command opening pump is not issued. Besides that, the detector can also easily find the exception. For example, detection based on event correlation [16] can easily identify the false sequence by analyzing the commands issued from the controller. Considering the mentioned two situations, a new attack model is described. It utilizes two attacks to simultaneously disrupt systems, which are manipulating maliciously aggregators of commands in the communication system to delay the disaggregation of some commands leading to the false sequential commands, and injecting false feedback sensory data by attacking the communication component leading to the false estimation. To illustrate the effectiveness of proposed attack model, we also discuss other possible attack paths and analyze whether the existing detection methods can identify attacks.

Our contributions are summarized as follows:

- We develop a simple and effective system model with intelligent control, which can describe the response from physical system handling command sequences issued by the controller.
- We describe a feasible false sequential command attack model, which is undetected and tempts the intelligent controller to issue successive commands leading to the false control.
- Combining with the system model, we discuss other possible attack paths that can generate false sequential commands and analyze whether these attack paths can be undetected by methods based on event correlation and based on false data evaluation [17]. The work demonstrates that the proposed model is more feasible.

The rest of paper is organized as follows. In Section 2, we review related work. We introduce preliminaries about communication system and detection methods in Section 3. In Section 4, we describe the system model and attack model. In Section 5, we analyze the feasibility of other possible attack paths. The case study is shown in Section 6. In Sections 7 and 8, we discuss the countermeasure and introduce the conclusion, respectively.

2. Related Work

In this section, we review state of the art on the cyber-physical attack and mainly introduce attacks related to false command control.

Currently, many security issues of CPSs have been discussed in previous research. In [5], false data injection was introduced. Attackers injected false data to disrupt the estimation of the system state. Especially, when some faults have occurred, false data injection leads to the concealment of false state and proper control can not be achieved in time. In [6], attackers described how to disrupt the smart grid by only injecting bad data. In [7,8], attackers used history data to replace the current data and mask the current state with going undetected. In [18], attackers collected information from the physical domain and revealed the information about the cyber domain. An example about 3D printers was described in detail. These prior works concentrated on how to input false data or steal data. However, the discussion about false command control has not been done.

In [9], the DOS attack in the power grid was proposed. The intruder can select any normal node as a puppet node and send false packets to infect other normal nodes. After that, the network communication bandwidth and node energy are consumed and DOS is caused. By using this attack, some commands may lose, which can lead to the false control. In [13,14], the time-delay attack was introduced, which illustrates that when the communication system is attacked and commands from controllers are delayed, some commands can not be executed in time and the disruption may occur. Besides that, false data injection in [19] can also cause false control from controllers. When false data is injected into the system, the false estimation may lead to false commands. Although the above research focused on how to cause false control from false commands, the false sequential command attack was not considered in detail. In [15], the false sequential command attack was first proposed and the authors built the attack model and showed the impact of attack by studying a case. However,

the model did not consider that the controller is intelligent and makes decisions based on the estimated state. Moreover, how to implement the attack and to escape from the detection was not considered.

3. Preliminaries

In this section, we first introduce how communication system transfers control commands and sensory data, and then introduce the preliminaries about attack methods and detection methods.

3.1. Data Transmission in the Communication System

Because of large scale and complex transaction, a typical hierarchical communication system is used in many large-scale CPSs [20–22], including many sub-controllers. Figure 3 shows the structure of communication system [23].

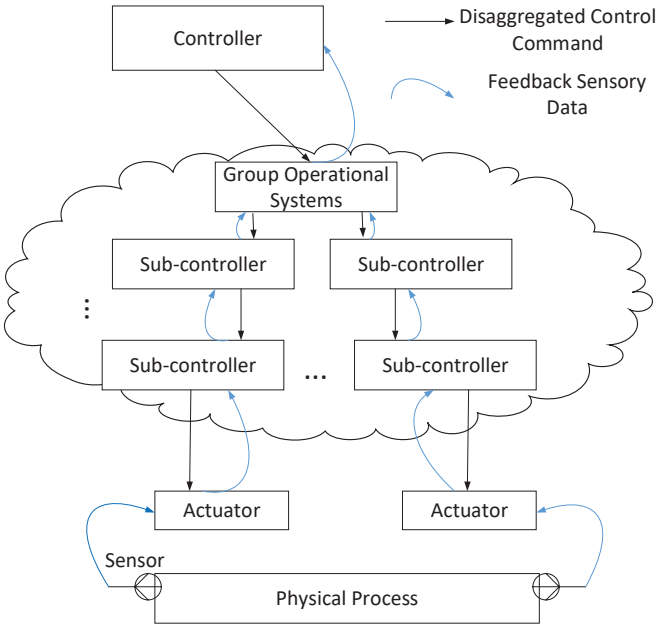


Figure 3. The structure of communication system.

The commands from the controller are disaggregated by the sub-controllers in every layer and are transferred to the next-tier sub-controllers. Min, B. et al. [20] describes an example about the power grid: assuming that the demand response (DR) load reduction of 70 MW is requested across the entire grid. Because DR capacities available on the subsystems are not equal due to their original capacity and the current state, this global command has to be disaggregated into a set of lower-level commands. This disaggregation process continues until local commands for endpoint field devices are generated and exercised.

During the regulation of control commands, sensory data is continuously fed back to the controller. The transmission path is contrary from the transmission path of commands. Sensory data is first transferred to actuators and then is sent to sub-controllers. At last, the sensory data is fed back to the controller. We neglect the time delay during the transmission process due to the transient time.

3.2. Attack Methods and Detection Methods

3.2.1. Attack Methods

We mainly pay attention to three attack entry points in Figure 1, including attacking controllers, attacking sub-controllers in communication system, and attacking sensors.

Attack 1: Manipulating the controller. Attackers have access to the controller and remotely manipulate the commands issued by the controller. For this attack, commands based on critical state estimation are ineffective and effective commands are generated based on the will of attackers.

Attack 2: Delaying the disaggregation of commands by attacking the sub-controllers. Attackers have access to some sub-controllers and manipulate the disaggregation of commands. When some commands are transferred to the sub-controllers, sub-controllers withhold these commands and disaggregate these commands base on attackers' input.

Attack 3: Injecting false data by attacking the sub-controllers. Attackers have access to some sub-controllers and can have authority to modify transferred feedback data. Sensory data from the sensors is replaced by the injected bad data from attackers.

Attack 4: Modifying the sensory data by capturing the sensors. Attackers can capture some sensors and inject false data into the sensors.

3.2.2. Detection Methods

We mainly focus on two countermeasures from the detector, including bad data evaluation based detection and event correlation based detection.

Countermeasure 1: Bad data evaluation based detection. The bad data evaluation is used to detect whether the sensory data is the normal response to the commands such as detecting bad data injection. The control commands and feedback sensory data are two input parameters. Every succeeding period of time, the system evaluates whether the current state is proper for the previous command. In this paper, we use the dissipativity-based fault detector [17] to represent the bad data evaluation. Once sensory data is modified and can not correspond to issued commands from the controllers, the detector will show an alarm. Although the method is effective to detect false data injection attack, when attackers inject the same data as the normal situation to conceal some faults and simultaneously launch other attacks such as false command injection attacks, the detection can not provide an alarm and a disaster may occur.

Countermeasure 2: Event correlation based detection [24,25]. Event correlation is used to identify the false command control. An event correlation refers to the correlation among multiple commands. For simplicity, we only use two commands c_i, c_j as an example to illustrate this method. When command c_j always occurs after command c_i is issued in the normal situation, sequence $< c_i, c_j >$ is seen as a correlation. When $< c_j, c_i >$ occurs, an alarm is shown.

4. System Model and Attack Model

In this section, we first describe the system model with intelligent control. Second, we propose a new false sequential command attack model based on delaying the disaggregation of commands and injecting false data.

4.1. System Model

We think that the controller is intelligent, which issues commands based on the critical state estimation. The system is modeled by a 7-tuple [15,26]:

$$P = \{C, AC, T, subT, S, R, S_{limit}\} \tag{1}$$

where

- $C = \{c_1, \ldots, c_m\}$ is a finite set of aggregated commands from the controller. c_i is the i_{th} kind of aggregated command. m denotes the number of commands.
- $AC = \{aC_1, \ldots, aC_m\}$ is a finite set of commands that are executed by actuators. $aC_i = \{ac_i(1), \ldots, ac_i(j), \ldots, ac_i(p)\}^T$ is disaggregated commands from the aggregated command c_i. $ac_i(j)$ is the command that is executed by the j_{th} actuator. p denotes the number of actuators.

- $T = \{t_1, \ldots, t_n\}$ is a finite set of time series. A time series is the measured values of one sensor with the change of time. $t_i = \{t_i(1), \ldots, t_i(k)\}^T$ means the time series from the i_{th} sensor. $t_i(l)$ denotes the measurement of the i_{th} sensor at time instant l.
- $subT = \{sub_1, \ldots, sub_{nd}\}$ is a finite set of time series, which is used to evaluate the critical state of the physical system. $subT$ is a subset of T. nd is the number of the time series.
- $S = \{s_1, \ldots, s_q\}$ is a finite set of states, where $s_i = \{a_1, \ldots, a_{nd}\}^T$ means one kind of system state, and is evaluated based on $subT$. The relationship can be described as

$$subT(k) = C_{matrix} \times S(k) \tag{2}$$

  where $C_{matrix} \in R^{nd \times nd}$ is the constant matrix and $S(k) \in S$ denotes the system state at time instant k. $subT(k) = \{sub_1(k), \ldots, sub_{nd}(k)\}^T$ where $sub_i(k)$ denotes the value of time series sub_i at time instant k. Equation (2) describes the critical state evaluation.
- $R = \{r_i, \ldots, r_{nr}\}$ is a finite set of relationship between the current state and commands from the controller, where, nr denotes the number of the relationships. $r_d = < s_i, c_j >$ denotes that when the state is s_i, the command from the controller is c_j. The state $S(k+1)$ at time instant $k+1$ is decided by $S(k)$ and $C(k)$, which can be described as

$$S(k+1) = A \times S(k) + B \times C(k) \tag{3}$$

  where $A \in R^{nd \times nd}$ and $B \in R^{nd \times 1}$ are constant matrices and $C(k)$ denotes the command from controller at time k.

  Equations (2) and (3) can describe the bad data evaluation. A, B and C are the coefficients in the control algorithm, which are determined by the specific system and can affect the result of detecting the system state. When one of two equations is violated, an alarm is shown [17].
- S_{limit} is a set of states, which is a subset of S. When the current state $S(t)$ is an element in set S_{limit}, a system fault occurs. In many systems, S_{limit} can be decided by the domain experts.

The model above is based on the assumption that information system and physical system have not yet been attacked, and all observed states and commands can be regarded as a representation of the normal system behavior.

4.2. Attack Model

The proposed method selects the place Y in Figure 1, the communication system, as the attack entry point.

We first assume that there exist some defects in the communication system and attackers can intrude into a part of components such as sub-controllers, which means attackers can remotely manipulate some components.

Attackers first control a part of sub-controllers in the communication system. Former command is not disaggregated until the latter command has been disaggregated from the corresponding sub-controller. During the process, if the former command is not disaggregated from the sub-controller, the system state may not be changed and the latter command is not issued from the controller. Thus, false data injection attack needs to be launched, which can deceive the controller to issue latter commands. Figure 4 describes the attack model. We only use two-tier controllers to describe the model. The detailed process is described as follows:

• Information collection

Before an attack is launched, related information needs to be collected to create better impact of attack. Especially, state sensing and sequence analysis are very important.

State sensing means that attackers need to sense the current state of the physical system. The current state can be evaluated based on the values of sensors. It is possible that high-skill

attackers can know the theory of critical state estimation, which means when attackers can collect sensory data, the state can be obtained.

Sequence analysis means that attackers need to analyze which sequence $< c_i, c_j >$ that satisfies (4) can achieve the state s_n defined in (5). When attackers find the kind of sequence, it means command sequence $< c_j, c_i >$ can disrupt the physical system.

$$\begin{cases} < s_i, c_i > \in R, \\ < s_j, c_j > \in R, \\ s_j = A \times s_i + B \times c_i, \\ s_m = A \times s_i + B \times c_j, \\ s_n = A \times s_m + B \times c_i. \end{cases} \tag{4}$$

$$s_n \in S_{limit} \tag{5}$$

• Time-delay attack

After attackers have selected a proper command sequence $< c_i, c_j >$, they try to control the sub-controller that will disaggregate the command c_i. When command c_i reaches the sub-controller, attackers manipulate the sub-controller to withhold the command for a while. The sub-controller that will disaggregate c_j is monitored. When attackers have known that command c_j is disaggregated, c_i can be disaggregated.

• False data injection

In this step, the attackers need to try to manipulate feedback values of sensory data. After a command c_k is executed, there exist many sensors whose measurements are changed. However, the system state is estimated based on $subT$ instead of T. Therefore, attackers only need to manipulate sub-controllers that transfer measurements of $sub_i \in sub(c_k)$, where $sub(c_k)$ means the set of time series whose values are changed after command c_k is executed.

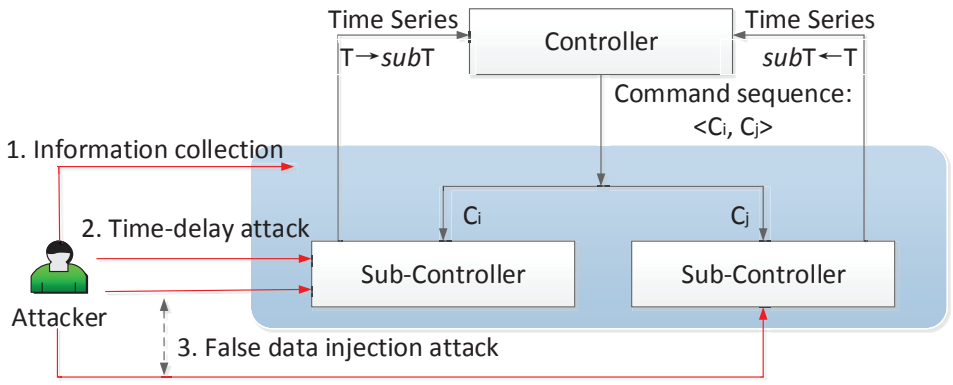


Figure 4. Attack model.

Attackers first try to get authority that can modify feedback data from sensors to the sub-controllers. After that, when the disaggregation of c_i is delayed, attackers need to inject bad data of time series $sub_i \in sub(c_i)$ into the sub-controller. The bad data $bad_i(c_i, k)$ of feedback data $sub_i(k)$ $(K1 \leq k \leq K2)$ where $K1$ means the time that c_i reaches the sub-controller and $K2$ denotes the time that c_i is executed), is equal to $sub_d(k)$ where sub_d and sub_i denote time series from the same sensor and $sub_d(k)$ can be computed by Equation (6). Previous such as [17] have proved that bad data satisfying Equation (6) is undetected by the bad data evaluation.

$$\{sub_1(k), \ldots, sub_{nd}(k)\}^T = C_{matrix} \times s_i \tag{6}$$

After the false data is injected, the controller issues the command c_j. After c_j is executed, $sub_j \in sub(c_j)$ is modified. The bad data $bad_j(c_j, k)$ of feedback data $sub_j(k)$ ($K3 \leq k \leq K2$ where $K3$ means the time that c_j is executed), is equal to $sub_e(k)$ where sub_e and sub_j denote the time series from the same sensor. $sub_e(k)$ can be computed by Equation (6).

The state transition of attack process can be described in (7).

$$\begin{cases} C(k) = c_i, \\ S(k) = s_m, \\ C(k+1) = c_j, \\ S(k+1) = s_n. \end{cases} \tag{7}$$

From the controller's point of view, command sequence $< c_i, c_j >$ is legal and event correlation based detection can not find an exception. Because of the injected bad data, the controller considers that the state is s_i at time k and the state is changed to s_j at time $k+1$. Bad data evaluation can not also find any exception. In fact, at time k , the real state is s_m. After c_i has been disaggregated and aC_i is executed by actuators, the state is changed to s_n and a fault occurs.

Figure 5a describes the normal situation. The proposed attack method tries to get a new situation that is shown in Figure 5b. The state refers to evaluated state based on critical state estimation. From the controller's point of view, the evaluated states and commands under the attacked situation are the same as the normal situation. Therefore, detection methods based on event correlation and bad data evaluation can not find the exception. At last, the sequence of commands is disordered before actuators execute disaggregated commands.

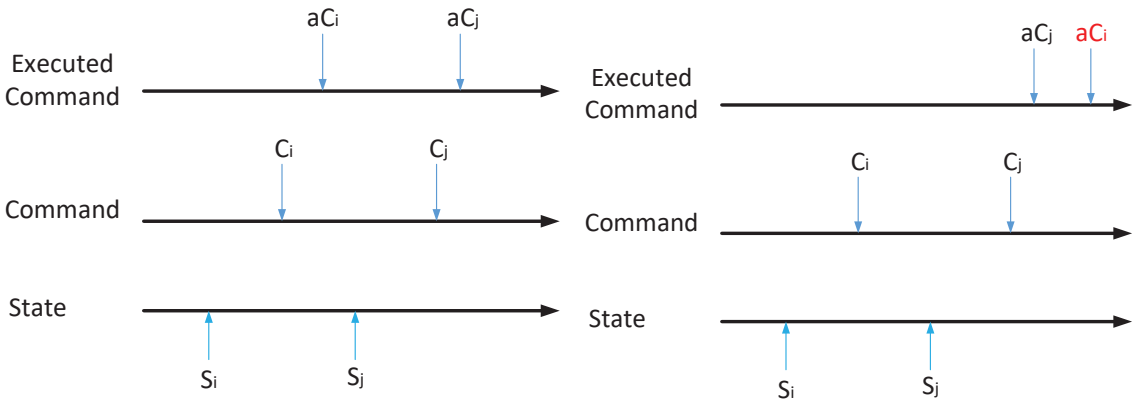


Figure 5. An example of the normal situation (**a**) vs. attack situation (**b**).

5. Analysis of other Paths

Combining the system model with the existing detection methods including event correlation and bad data evaluation, we discuss the feasibility of other attack paths to launch false sequential command attacks. Our discussion illustrate that the proposed model is more feasible.

As described in Figure 1, there exist three entry points including place X, place Y, and place Z that can disrupt the system. We mainly focus on attack methods including manipulating the controller, delaying the disaggregation of commands, injecting false data by attacking sub-controllers, and modifying the sensory data by capturing the sensors. We first analyze situations that a single entry

point is attacked, and then discuss the situations of multiple entry points. Normal sequence $< c_i, c_j >$ satisfies (4) and (5).

5.1. Attack Based on a Single Entry Point

• Place Z in Figure 1 as the attack entry point

When attackers can maliciously manipulate the controller and hope to issue false sequential commands $< c_j, c_i >$, the physical process can be described in (8).

$$\begin{cases} S(k) = s_i, \\ C(k) = c_j, \\ S(k+1) = s_m = A \times s_i + B \times c_j, \\ C(k+1) = c_i. \end{cases} \tag{8}$$

If attackers execute these operations, bad data evaluation can find the exception because $< s_i, c_j >$ is not an element from set R. Moreover, event correlation also can find that $< c_j, c_i >$ is an exceptional sequence.

Therefore, only attacking the entry point Z can not realize the false sequential command attack.

• Place X in Figure 1 as the attack entry point

Attackers need to capture sensors and inject false data to construct the false state leading to false sequential commands $< c_j, c_i >$. Attackers first should tell the controller that the current state is s_j, and then falsify the next state s_i, which seems to be feasible. The mentioned process can be described in (9).

$$\begin{cases} S(k) = s_j, \\ C(k) = c_j, \\ S(k+1) = s_i, \\ C(k+1) = c_i. \end{cases} \tag{9}$$

However, bad data evaluation can find that the executed command c_j leads to the state s_i, but $s_i = A \times s_j + B \times c_j$ is wrong. An alarm is shown, which illustrates that only capturing the sensors to modify sensory data can not obtain effective impact of the false sequential command attack.

• Place Y in Figure 1 as the attack entry point

Besides the proposed attack model in Section 4, attackers can only launch time-delay attacks (delaying the disaggregation of commands by attacking the sub-controllers) or inject bad data by attacking the sub-controller to disorder the command sequence.

When attackers only delay the disaggregation of command c_i, the state s_i is not changed. (10) can describe the process. The controller can not issue command c_j. The false command sequence can not be achieved.

$$\begin{cases} S(k) = s_i, \\ C(k) = c_i, \\ S(k+1) = s_i, \\ C(k+1) = c_i. \end{cases} \tag{10}$$

When bad data is injected into the communication system, attackers can get the same result as attacking sensors in place X.

The above two situations can not get the effective impact of false sequential command attacks.

5.2. Attack Based on Multiple Entry Points

• Places Y and Z as attack entry points or places X and Z as attack entry points

For this kind of path, attackers modify command sequences by intruding into the controller and falsify bad data about sensors by capturing the sensors or attacking the sub-controllers.

When attackers manipulate the controller to issue false sequence $< c_j, c_i >$, feedback data at time $k+1$ is modified and the evaluated state is s_i at time $k+1$. The real state transition is described in (11).

$$\begin{cases} S(k) = s_i, \\ C(k) = c_j, \\ S(k+1) = s_j, \\ C(k+1) = c_i. \end{cases} \tag{11}$$

For the attack, event correlation can easily find an exception because $< c_j, c_i >$ is illegal.

When bad data is first injected, manipulating the controller becomes useless. The situation is same as attacking the single place X.

• Places Y and X as attack entry points

Attackers manipulate the command sequence by intruding into the communication system and falsify data by capturing sensors. The above process is similar to the proposed method, and can get the same state transition. Different from the proposed method, the attackers need to capture sensors. This method may need to control fewer sub-controllers in the communication system. Attackers need to select the proper path based on the real ability.

Based on the above analysis, we can know only when attackers simultaneously disorder the command sequence and contaminate feedback data, can the false command sequence be achieved, while ensuring that the order of commands is not false when they are issued from the controller.

6. Case Study

In this section, we study a case about tank system to demonstrate the feasibility of attack process and the impact of attack.

6.1. Scenario

Inspired by [15,27], we construct two three-tank systems with the same function, as shown in Figure 6. Every time, the controller issues the same command into sub-controller 1 and sub-controller 2. Next, we describe a three-tank sub-system to illustrate the control process.

The three-tank system provides liquid C that is produced in tank $T13$. The liquid C can be achieved by the neutralization process of ingredient A and ingredient B. The ratio of ingredient A to ingredient B is 1. The error is allowed within ten percent. There are 6 commands and three sensors as shown in Table 1. Sensor 1 measures the values of ingredient A in tank $T11$. Sensor 2 measures the changes of ingredient B in tank $T12$ and Sensor 3 senses the changes of liquid C in tank $T13$. When the pump or valve is opened, the liquid A and liquid B flow out from the components by 3 mL/s, and the liquid C flows out from the tank $T13$ by 6 mL/s.

The control process and state description are shown in Tables 2 and 3. The command sequence $\{P1o, P1f, P2o, P2f, V11o, V11f\}$ is repetitively issued. The controller based on the critical state estimation issues the next command based on the current state.

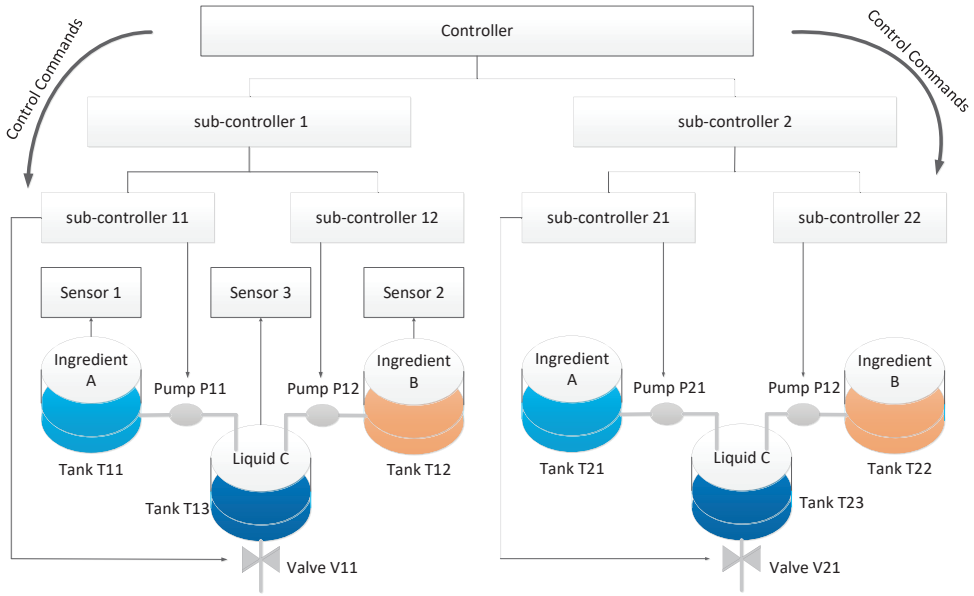


Figure 6. Structure of tank system. Adapted with permission from [15], Copyright Elsevier, 2016.

Table 1. Description of data.

Command/Time Series	Description
$P1o/P1f$	Switch on/off Pump 11 and Pump 21
$P2o/P2f$	Switch on/off Pump 12 and Pump 22
$V11o/V11c$	Open/Close Valve
$T1$	Measurements of Sensor 1
$T2$	Measurements of Sensor 2
$T3$	Measurements of Sensor 3

Table 2. Control Process where After TC_i denotes the time that C_i has been issued from the controller.

Control	Constraints	Step
$P1o$	S_0	Step 1
$P1f$	S_1 and After $T(P1o) \geq 1$ min	Step 2
$P2o$	S_2 and After $T(P1f) \geq 1$ min	Step 3
$P2f$	S_3 and After $T(P2o) \geq 1$ min	Step 4
$V11o$	S_4 and After $T(P2f) \geq 1$ min	Step 5
$V11f$	S_5 and After $T(Vllo) \geq 2$ min	Step 6

Table 3. Description of states.

State	Description
S_0	$T3(z) = 0$
S_1	$T1(z) = T1(z-1) - 3$
S_2	$180 - 18 \leq T3(z) \leq 180 + 18$ and $T1(z) = T1(z-1)$
S_3	$T2(z) = T2(z-1) - 3$
S_4	$360 - 36 \leq T3(z) \leq 360 + 36$ and $T2(z) = T2(z-1)$
S_5	$T3(z) - T3(z-1) = 6$

The values of time series $T1$, $T2$, and $T3$ under the normal situation are shown in Figure 7.

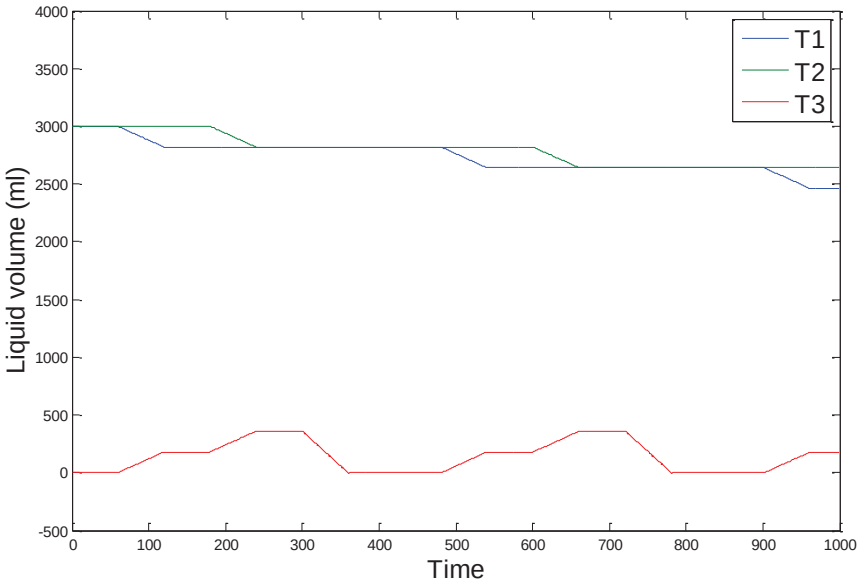


Figure 7. The values of sensors with the change of time under the normal situation.

6.2. Attack Cases

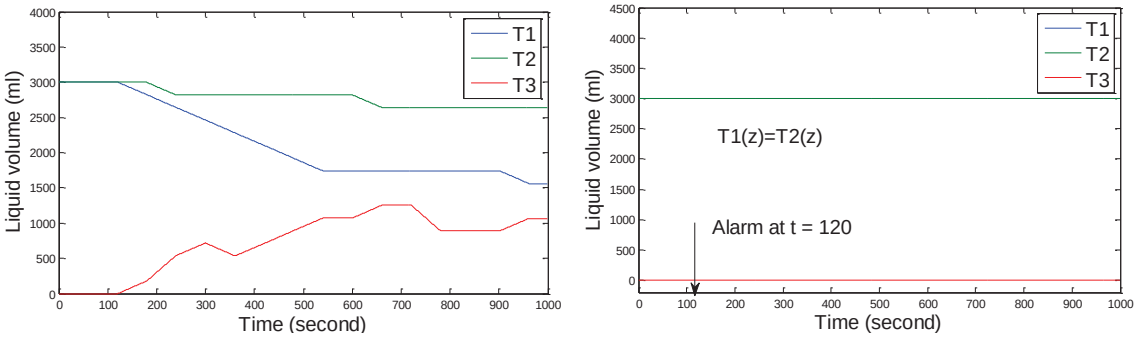
We assume that attackers have controlled the sub-controller 11 and sub-controller 12. Four attack cases are described as follows:

- Case 1: Attackers delay the disaggregation of $P1o$ until the command $P1f$ has been disaggregated. $P1f$ is disaggregated at $t = 120$ s, and $P1o$ is disaggregated at $t = 121$ s. The attacker injects false data about $T1$ and $T3$ to keep the same as Figure 7.
- Case 2: Attackers only delay the disaggregation of $P1o$ until the command $P1f$ has been disaggregated. The command $P1f$ is injected into the controller and disaggregated at $t = 120$ s, and $P1o$ is disaggregated at $t = 121$ s.
- Case 3: Attackers delay the disaggregation of $V11o$ until the command $V11f$ has been disaggregated. The command $V11f$ is disaggregated at $t = 361$ s and $V11o$ is disaggregated at $t = 370$ s. The attacker injects false data about $T3$ to keep the same as Figure 7.
- Case 4: Attackers only delay the disaggregation $V11o$ until $t = 370$ s. Under the normal situation, $V11f$ is issued at $t = 361$ s.

6.3. Impact of Attack

Figure 8a shows the real values of $T1$, $T2$, and $T3$ under attack case 1. We can observe that when sequence $< P1o, P1f >$ is changed to $< P1f, P1o >$, ingredient A is increased continuously. Until the command $V11o$ is executed at $t = 300$ s, the ratio of ingredient A to ingredient B is not 1. Moreover, because ingredient A is injected from $t = 61$ s to $t = 541$ s, there exists ingredient A in tank $T13$ when the second cycle begins, which means although the command sequence of the second cycle is normal, if attackers still inject false data to conceal the real state, the false liquid in tank $T13$ is still obtained.

Figure 8b shows the real values of $T1$, $T2$, and $T3$ under attack case 2. When the disaggregation of $P1o$ is delayed, the state is still S_0. Until the command $P1f$ is disaggregated at $t = 120$ s, the state transition has an exception and an alarm occurs. During the process, the liquid of Tank $T13$ is 0 mL. Attacks can be identified and there is no economic loss.



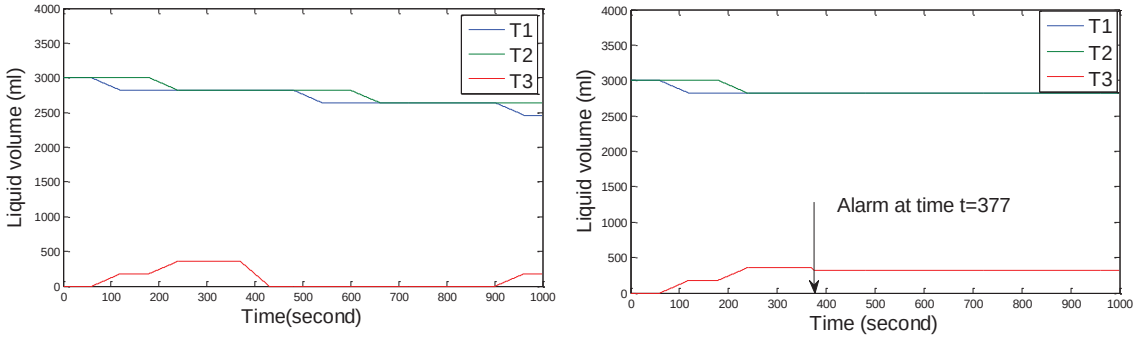
(**a**) Case 1 (**b**) Case 2

Figure 8. The real values of sensors under false sequence $< P1f, P1o >$.

Comparing the above two cases, we can say that the combination of false data injection and time delay attacks is necessary for the kind of attack. Only changing the order of command sequences can not cause the disruption.

Figure 9a describes the real values of $T1$, $T2$, and $T3$ under attack case 3. We can clearly see that when sequence $< V11o, V11f >$ is disordered at the first circle, the liquid C can be achieved after the first circle. Different from the normal situation, the time of getting liquid C is a little later. However, at the second circle, the fault occurs. Although A and B output normally from the tanks, tank $T13$ does not store liquid. Until the third circle, the process is normal. During the attack, attackers inject false data about $T13$ from $t = 240$ s to $t = 840$ s. Comparing the result with case 1, we can find that the different levels of impact are achieved by different false command sequences.

Figure 9b shows the real values of $T1$, $T2$, and $T3$ under attack case 4. We can observe that sequence $< V11o, V11f >$ can not be disordered at the first circle and an alarm is shown because command $V11f$ is issued at $t = 377$ s, which causes a false state transition. Comparing with other cases, we can say that how to deceive the controller to issue the next command must be considered. The above case demonstrates that the proposed method is feasible to launch false sequential command attacks.



(**a**) Case 3 (**b**) Case 4

Figure 9. The real values of sensors under false sequence $< V11o, V11f >$.

7. Discussion of Countermeasure

The detector using event correlation can not identify the exceptions because false sequences are not collected. If defenders can collect the false sequences, event correlation can find these exceptions. Therefore, we propose a two-tier event correlation based detection method. Detector collects commands from two places, including output of controllers and output of sub-controllers. There exists a fixed correlation between command sequence $< c_i, c_j >$ from the controller and disaggregated command

sequence $< aC_i, aC_j >$. Detector can use the correlation to find false sequential command attack. For example, as shown in Figure 10, under the normal situation, command c_i is issued from the controller at time t and aC_i is issued from sub-controllers at time $t + d_i$. After aC_i is executed, c_j is issued from the controller at time k and aC_i is issued from sub-controllers at time $k + d_j$. When the false sequence command attack occurs, the detector will obtain that aCi is issued after command aC_j occurs. Sequence $< ci, cj >$ and sequence $< aC_j, aC_i >$ are not correlated and anomalies are alarmed. In the future, we will study two-tier correlation based detection in depth.

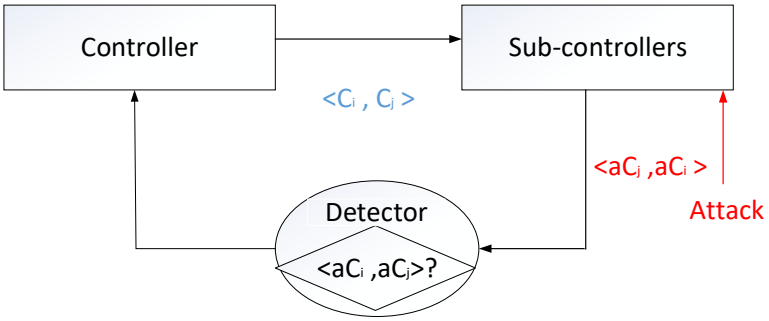


Figure 10. Countermeasure.

8. Conclusions

In this paper, we mainly focus on the false sequential command attack. Different from the previous research, we consider that the system with detectors can effectively identify attacks. It means that only modifying the order of commands issued from the controller is ineffective. We propose a feasible attack model, which uses time-delay attacks to disorder the command sequence and bad data injection to interfere with the estimation of the system state. The attack can be undetected by the existing detection methods. We also analyze other possible attack paths. The work demonstrates that the proposed model is more feasible. A case study is given to demonstrate that the described attack model is effective and feasible to disrupt the physical system. Finally, we discuss the possible countermeasure. However, there are some limitations in this model. For example, we did not consider the impact of measurement errors. It is the key to solve the limitations in further research.

Author Contributions: Conceptualization, Z.Y. and Y. X.; Methodology, P. X.; Software, B.W. and T.D.; Validation, Z.Y. and B.W.; Formal Analysis, Z.Y. and Y.X.; Data Curation, P.X.; Writing—Original Draft Preparation, Z.Y.; Writing—Review & Editing, P.X. and Y.X.; Supervision, Z.Y.; Project Administration, Y.X.

Funding: This research was funded by the Natural Science Foundation of Hunan Province under Grant No. 2017JJ2292; Science and Technology Key Projects of Hunan Province under Grant No. 2016JC2012; Outstanding Youth Research Project of Provincial Education Department of Hunan under Grant No. 17B030; Science and Technology Planning Project of Changsha under Grant No. K1705018, ZD1601042 and K1705031.

Acknowledgments: The authors sincerely thank the anonymous referees for their invaluable suggestions that have led to the present improved version of the original manuscript.

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

References

1. Pagliari, L.; Mirandola, R.; Trubiani, C. Multi-modeling Approach to Performance Engineering of Cyber-Physical Systems Design. In Proceedings of the 2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS), Fukuoka, Japan, 5–8 November 2017; pp. 142–145.
2. Tian, J.; Tan, R.; Guan, X.; Liu, T. Enhanced Hidden Moving Target Defense in Smart Grids. *IEEE Trans. Smart Grid* **2018**. [CrossRef]
3. Xun, P.; Zhu, P.D.; Maharjan, S.; Cui, P.S. Successive direct load altering attack in smart grid. *Comput. Secur.* **2018**, *77*, 79–93. [CrossRef]

4. Liang, G.; Weller, S.R.; Zhao, J.; Luo, F.; Dong, Z.Y. The 2015 Ukraine Blackout: Implications for False Data Injection Attacks. *IEEE Trans. Power Syst.* **2017**, *32*, 3317–3318. [CrossRef]
5. Liang, J.; Sankar, L.; Kosut, O. Vulnerability analysis and consequences of false data injection attack on power system state estimation. *IEEE Trans. Power Syst.* **2016**, *31*, 3864–3872. [CrossRef]
6. Amini, S.; Mohsenian-Rad, H.; Pasqualetti, F. Dynamic load altering attacks in smart grid. In Proceedings of the 2015 IEEE Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 18–20 February 2015; pp. 1–5.
7. Wang, J.; Tu, W.; Hui, L.C.K.; Yiu, S.M.; Wang, E.K. Detecting Time Synchronization Attacks in Cyber-Physical Systems with Machine Learning Techniques. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 2246–2251.
8. Garcia, L.A.; Brasser, F.; Cintuglu, M.H.; Zonouz, S.A. Hey, My Malware Knows Physics! Attacking PLCs with Physical Model Aware Rootkit. In Proceedings of the 2017 Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 26 February–1 March 2017; pp. 1–15.
9. Lin, H.; Slagell, A.; Kalbarczyk, Z.; Sauer, P.W.; Iyer, R.K. Semantic Security Analysis of SCADA Networks to Detect Malicious Control Commands in Power Grids. In Proceedings of the First ACM Workshop on Smart Energy Grid Security, Berlin, Germany, 8 November 2013; pp. 29–34.
10. Yuan, H.; Xia, Y. Resilient strategy design for cyber-physical system under DoS attack over a multi-channel framework. *Inf. Sci.* **2018**, *454*, 312–327. [CrossRef]
11. Yi, P.; Zhu, T.; Zhang, Q.; Wu, Y.; Li, J. A denial of service attack in advanced metering infrastructure network. In Proceedings of the 2014 IEEE International Conference (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 1029–1034.
12. Zhang, H.; Zheng, W.X. Denial-of-Service Power Dispatch against Linear Quadratic Control via a Fading Channel. *IEEE Trans. Autom. Control* **2018**, *63*, 3032–3039. [CrossRef]
13. Sargolzaei, A.; Yen, K.K.; Abdelghani, M.N.; Sargolzaei, S.; Carbunar, B. Resilient Design of Networked Control Systems Under Time Delay Switch Attacks, Application in Smart Grid. *IEEE Access* **2017**, *5*, 15901–15912. [CrossRef]
14. Sargolzaei, A.; Yen, K.K.; Abdelghani, M.N. Preventing Time-Delay Switch Attack on Load Frequency Control in Distributed Power Systems. *IEEE Trans. Smart Grid* **2016**, *7*, 1176–1185. [CrossRef]
15. Li, W.; Xie, L.; Deng, Z.; Wang, Z. False sequential logic attack on scada system and its physical impact analysis. *Comput. Secur.* **2016**, *58*, 149–159. [CrossRef]
16. Han, Y.; Zhu, M.; Liu, C. A Service-Oriented Approach to Modeling and Reusing Event Correlations. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; pp. 498–507.
17. Vu, Q.D.; Tan, R.; Yau, D.K.Y. On Applying Fault Detectors against False Data Injection Attacks in Cyber-Physical Control Systems. In Proceedings of the IEEE INFOCOM 2016—35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
18. Abdullah, A.F.M.; Rokka, C.S.; Arquimedes, C.; Jiang, W. Acoustic Side-channel Attacks on Additive Manufacturing Systems. In Proceedings of the 7th International Conference on Cyber-Physical Systems, Vienna, Austria, 11–14 April 2016; p. 19.
19. Hu, L.; Wang, Z.; Han, Q.L.; Liu, X. State estimation under false data injection attacks: Security analysis and system protection. *Automatica* **2018**, *87*, 176–183. [CrossRef]
20. Min, B.; Varadharajan, V. Cascading attacks against smart grid using control command disaggregation and services. In Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, 4–8 April 2016; pp. 2142–2147.
21. Remmersmann, T.; Schade, U.; Schlick, C. Supervisory control of multi-robot systems by disaggregation and scheduling of quasi-natural language commands. In Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, Korea, 14–17 October 2012; pp. 315–320.
22. Taft, J.D. Control Command Disaggregation and Distribution within a Utility Grid. U.S. Patent Application No 13/484,042, 30 May 2012.
23. Sargolzaei, A.; Yen, K.; Abdelghani, M.N. Delayed inputs attack on load frequency control in smart grid. In Proceedings of the ISGT 2014 IEEE, Washington, DC, USA, 19–22 February 2014; pp. 1–5.
24. Shiva, S.; Dharam, R.; Shandilya, V. Runtime Monitors as Sensors of Security Systems. In Proceedings of the 23rd IASTED International Conference, Dallas, TX, USA, 14–16 December 2011.

25. Pan, S.; Morris, T.; Adhikari, U. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Trans. Smart Grid* **2015**, *6*, 3104–3113. [CrossRef]
26. Shandilya, V.; Simmons, C.B.; Shiva, S. Use of attack graphs in security systems. *J. Comput. Netw. Commun.* **2014**, *2014*. [CrossRef]
27. Renganathan, K.; Bhaskar, V. Observer based on-line fault diagnosis of continuous systems modeled as Petri nets. *ISA Trans.* **2010**, *49*, 587–595. [CrossRef] [PubMed]