available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/ijcip

Understanding the physical and economic consequences of attacks on control systems

Yu-Lun Huang[c,*], *Alvaro A. Cárdenas*[a], *Saurabh Amin*[b], *Zong-Syun Lin*[c], *Hsin-Yi Tsai*[c], *Shankar Sastry*[a]

[a] *Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California 94720, USA*
[b] *Department of Civil and Environmental Engineering, University of California, Berkeley, California 94720, USA*
[c] *Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, 30010, Taiwan*

ARTICLE INFO

Article history:
Received 11 June 2009
Accepted 11 June 2009

Keywords:
Control systems
Integrity attacks
Denial-of-service attacks
Consequences

ABSTRACT

This paper describes an approach for developing threat models for attacks on control systems. These models are useful for analyzing the actions taken by an attacker who gains access to control system assets and for evaluating the effects of the attacker's actions on the physical process being controlled. The paper proposes models for integrity attacks and denial-of-service (DoS) attacks, and evaluates the physical and economic consequences of the attacks on a chemical reactor system. The analysis reveals two important points. First, a DoS attack does not have a significant effect when the reactor is in the steady state; however, combining the DoS attack with a relatively innocuous integrity attack rapidly causes the reactor to move to an unsafe state. Second, an attack that seeks to increase the operational cost of the chemical reactor involves a radically different strategy than an attack on plant safety (i.e., one that seeks to shut down the reactor or cause an explosion).

1. Introduction

Control systems are computer-based systems used to monitor and control physical processes. They are usually composed of a set of networked devices such as sensors, actuators, controllers, and communication devices.

Control systems and networks are essential to monitoring and controlling many critical infrastructure assets (e.g., electric power distribution, water treatment, and transportation management) and industrial plants (e.g., those used for manufacturing chemicals, pharmaceuticals, and food products). Most of these infrastructures are safety-critical – an attack can impact public health, the environment, the economy, and even lead to the loss of human life.

Control systems are becoming more complex and interdependent and, therefore, more vulnerable. The increased risk of computer attacks has led to numerous investigations of control system security (see, e.g., [1–11]). Most of the technical solutions involve extensions and improvements to traditional information technology (IT) mechanisms. However, very few solutions consider the interactions between security and the physical processes being controlled. In particular, researchers have not considered how attacks affect the estimation and control algorithms that regulate physical systems, and, ultimately, how the attacks affect the physical environment.

The goal of this paper is to initiate the development of new threat models for control systems. We argue that a threat assessment must include an analysis of how attacks on control systems can affect the physical environment in order to: (i) understand the consequences of attacks, (ii) estimate the possible losses, (iii) estimate the response time required by defenders, and (iv) identify the most cost-effective defenses.

* *Corresponding author.* Tel.: +886 3 5131476.
E-mail address: ylhuang@cn.nctu.edu.tw (Y.-L. Huang).

1874-5482/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijcip.2009.06.001

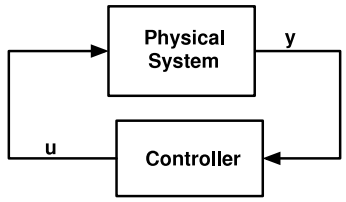

Fig. 1 – Control system abstraction.

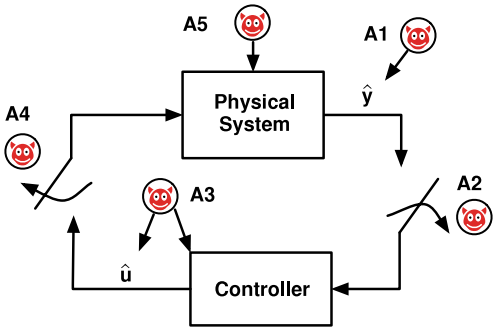

Fig. 2 – Attacks on control systems.

The paper is organized as follows. The next section, Section 2, focuses on formal models of cyber attacks in control systems. Section 3 describes the experimental setup and analyzes the experimental results. The final section, Section 4, summarizes our conclusions and highlights areas for future research.

2. Modeling Attacks

This section defines the control system abstraction and formally models integrity and denial of service (DoS) attacks.

2.1. *Notation*

A control system is composed of sensors, controllers, actuators, and the physical system (plant). Sensors monitor the physical system and send measurements to a controller. The controller sends control signals to actuators. Upon receiving a control signal, an actuator performs a physical action (e.g., opening a valve). Fig. 1 clarifies the relationships between the physical system, sensor signals (y), the controller, and control signals (u).

The following notation is used to formally model attacks on control systems.

- Time (t): The term t denotes an instant of time. A process runs from $t = 0$ to $t = T$.
- Sensor Measurement ($y_i(t)$): The term $y_i(t)$ denotes the value measured by sensor i at time t. Note that, $\forall i, t$, $y_i(t) \in \mathcal{Y}$, where $\mathcal{Y} = [y_i^{min}, y_i^{max}]$ (y_i^{min} and y_i^{max}) are the reasonable minimum and maximum values representing the plant state, respectively. Also, $Y = [y_1, y_2, \ldots, y_n]^T$, where n is the number of sensors.
- Manipulated Variable ($u_i(t)$): The term $u_i(t)$ denotes the output of controller i at time t. Note that, $\forall i, t$, $u_i(t) \in \mathcal{U}_i$, where $\mathcal{U}_i = [u_i^{min}, u_i^{max}]$ is the allowable range of controller output values.
- Attack Duration ($\mathcal{T}_a$): The term $\mathcal{T}_a$ denotes the duration of an attack. An attack starts at $t = t_s$ and ends at $t = t_e$. Note that $\mathcal{T}_a = [t_s, t_e]$.

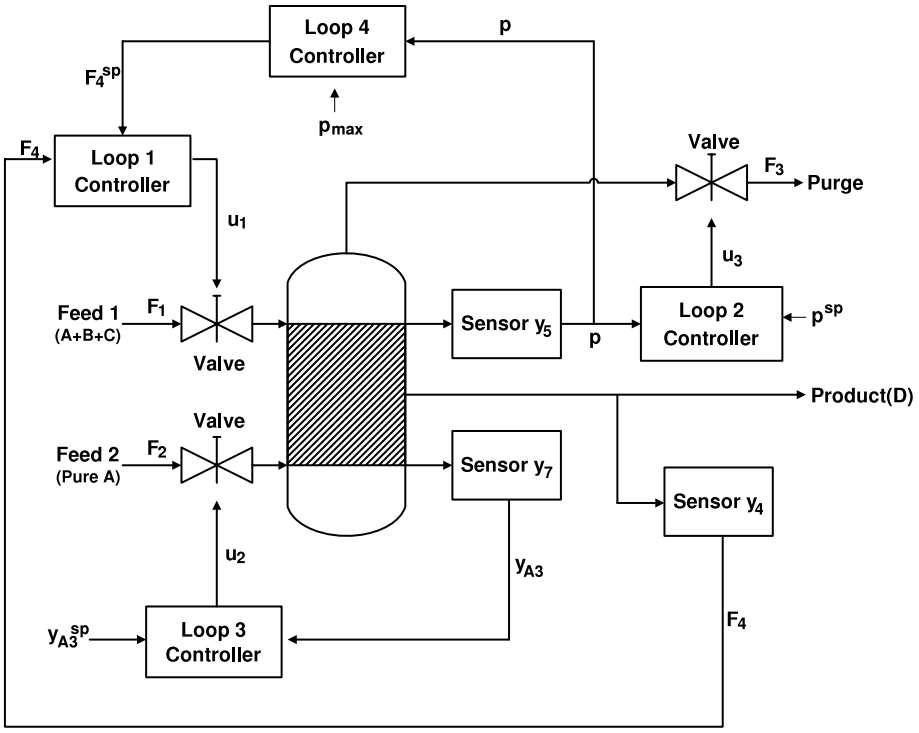

Fig. 3 – Chemical plant.

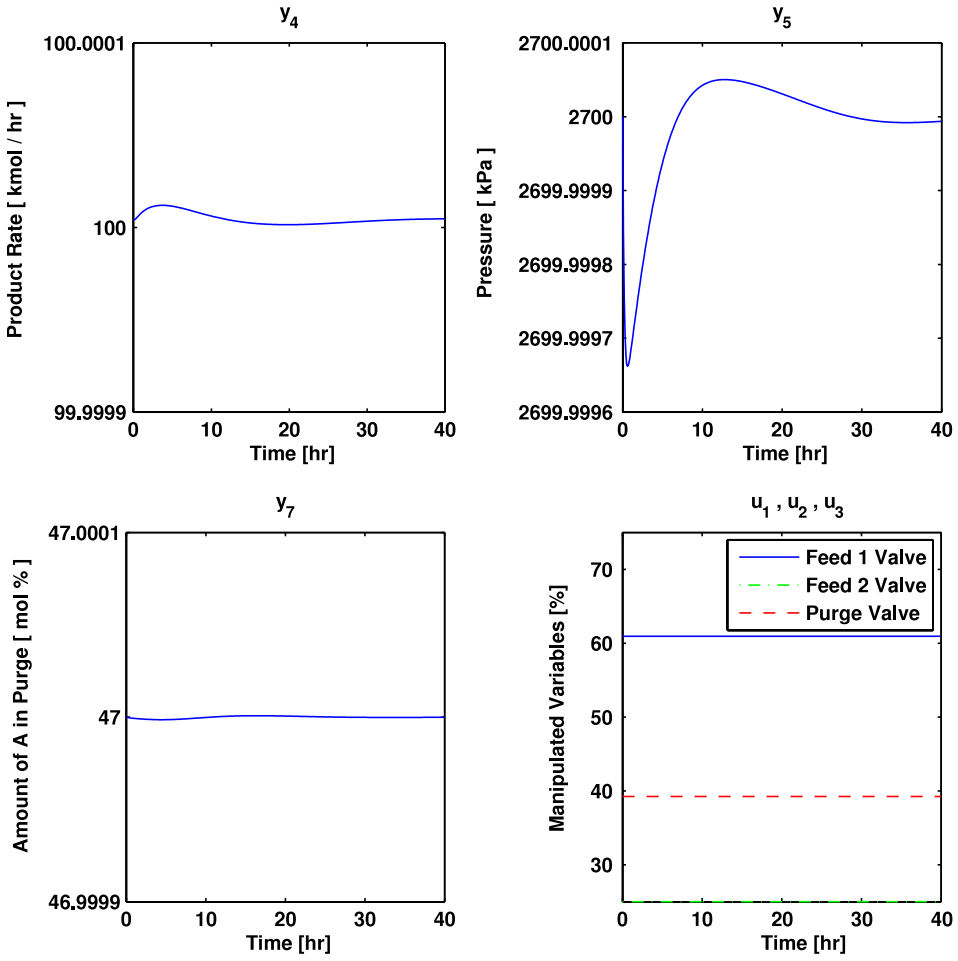

Fig. 4 – Plant outputs without noise.

Fig. 2 identifies several attacks on control systems. A1 and A3 correspond to *integrity attacks*, where the adversary sends false information $\hat{y} \neq y$ or $\hat{u} \neq u$ from (one or more) sensors or controllers. The false information may be an incorrect measurement, an incorrect time when the measurement was observed, or an incorrect sender identifier. The adversary can launch these attacks by obtaining the secret keys used by the devices or by compromising sensors (A1) or controllers (A3). We assume that each device is uniquely authenticated. Therefore, an attacker who compromises the secret key of a device is able to impersonate only that device.

A2 and A4 correspond to *DoS attacks*, where the adversary prevents the controller from receiving sensor measurements or prevents actuators from receiving control commands. The adversary can launch a DoS attack by jamming communication channels, compromising devices and preventing them from sending data, attacking routing protocols, or flooding the network.

A5 corresponds to a *direct attack* against actuators or an *external physical attack* on the plant. From an algorithmic perspective, it is not possible to defend against such attacks (aside from detecting them). Therefore, significant efforts must be implemented to deter and/or prevent attacks against physical systems (e.g., by implementing physical security controls).

2.2. *Modeling integrity attacks*

A successful integrity attack on sensor i modifies the real sensor signal, causing the input to the control function u to be changed from y to $\hat{y}$. In an integrity attack, the adversary sends a value $\hat{y}$ or $\hat{u}$ to a sensor or actuator based on the information available to the adversary.

In an effort to develop a systematic – and trackable – treatment of attack strategies, we propose the investigation of *max attacks*, *min attacks*, *scaling attacks*, and *additive attacks*. We assume that all these attacks lie within $\mathcal{U}_i$ and $\mathcal{Y}_i$. Note that signals outside this range are easily detected by fault-tolerant algorithms.

The following attacks can be launched against sensors:

- Min and Max Attacks:

$$\hat{y}_i^{\min}(t) = \begin{cases} y_i(t) & \text{for } t \notin \mathcal{T}_a \\ y_i^{\min} & \text{for } t \in \mathcal{T}_a, \end{cases}$$

and

$$\hat{y}_i^{\max}(t) = \begin{cases} y_i(t) & \text{for } t \notin \mathcal{T}_a \\ y_i^{\max} & \text{for } t \in \mathcal{T}_a. \end{cases}$$

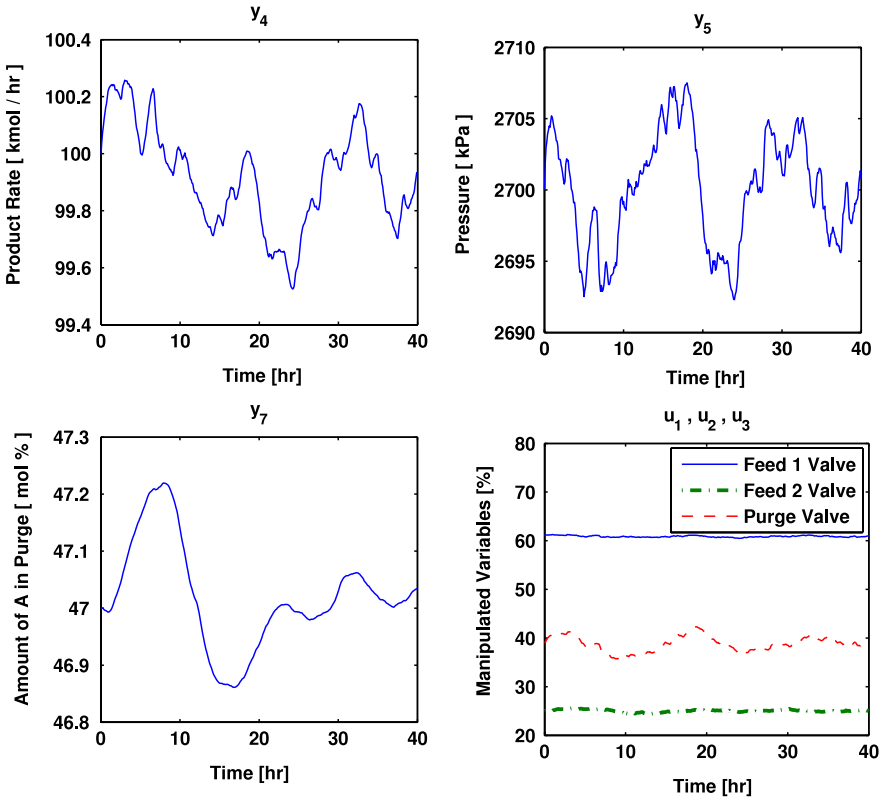

Fig. 5 – Plant outputs with Gaussian noise.

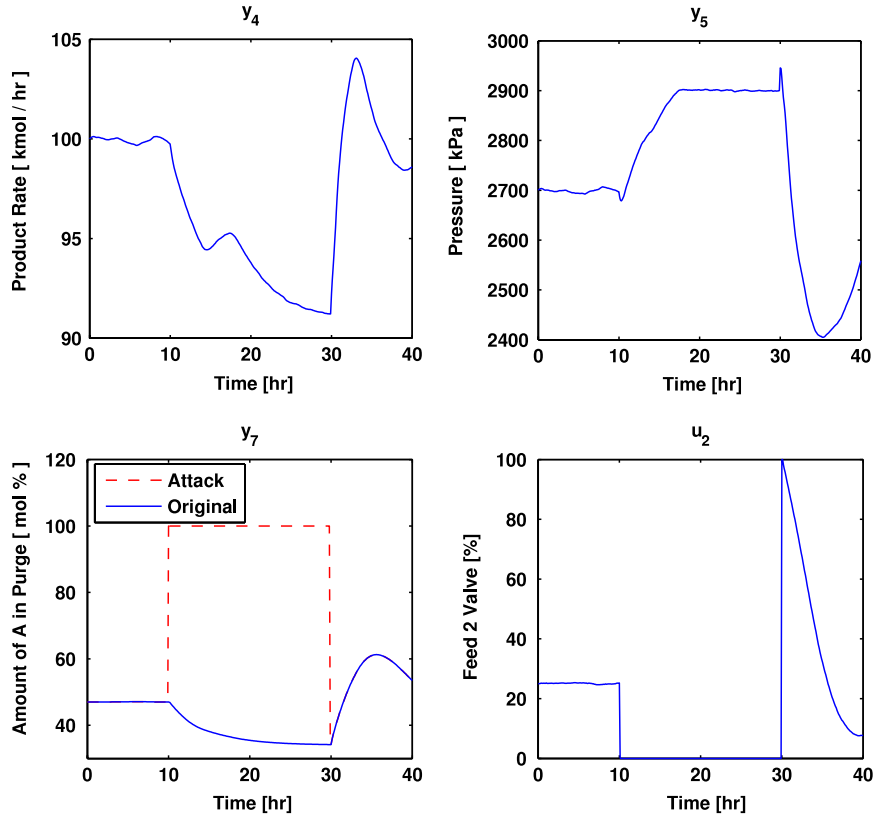

Fig. 6 – Integrity attack y_7^{max} from $t = 0$ to $t = 30$.

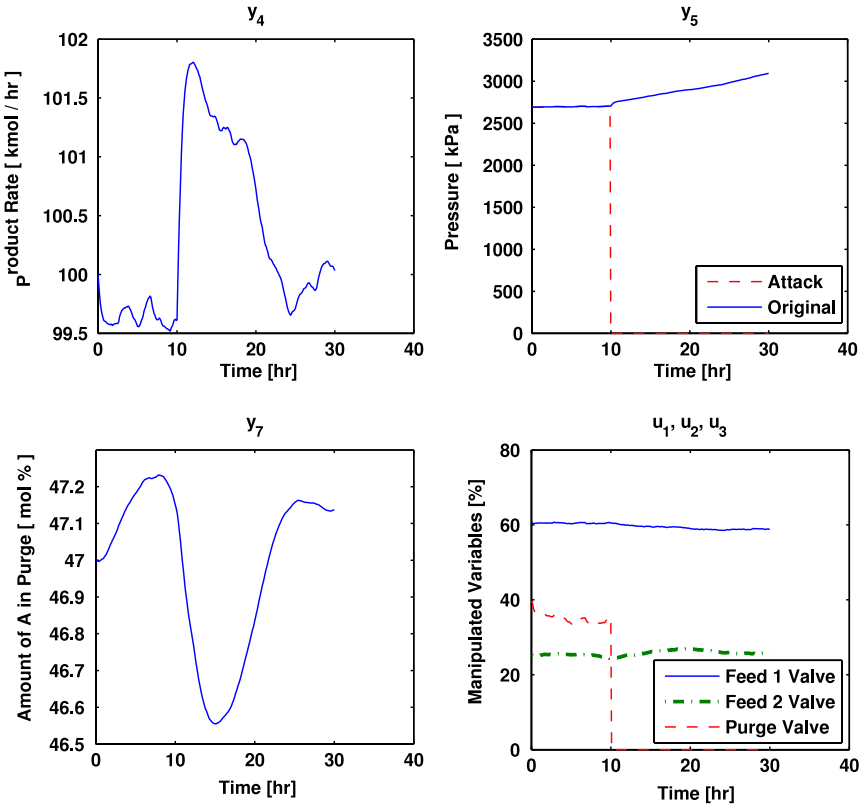

Fig. 7 – Integrity attack $y_5^{\min}$ from $t = 0$ to $t = 30$.

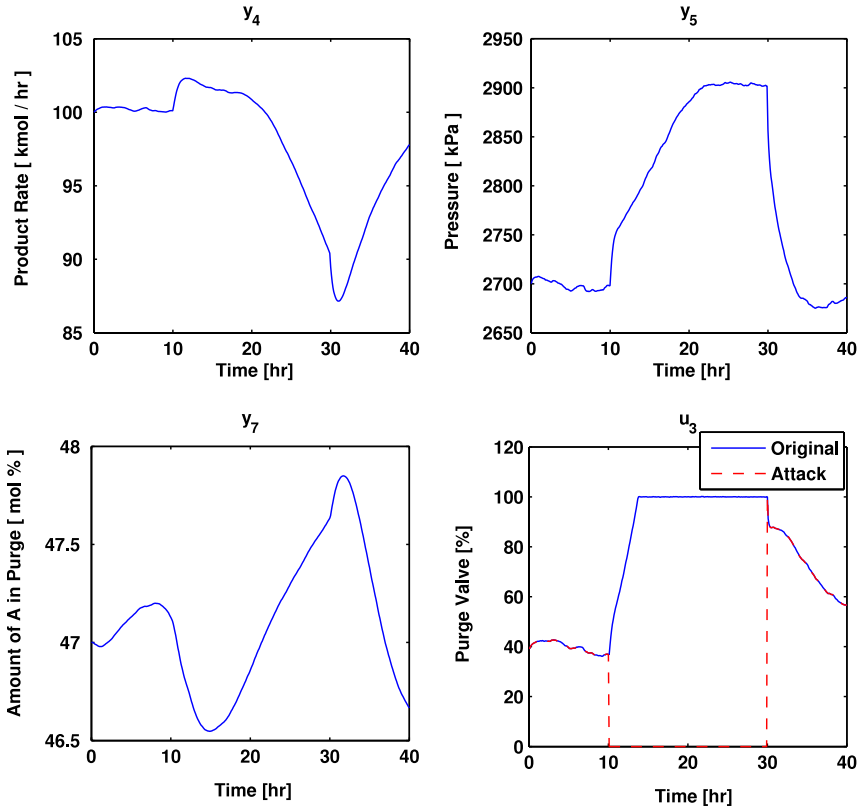

Fig. 8 – Integrity attack $u_3^{\min}$ from $t = 0$ to $t = 30$.

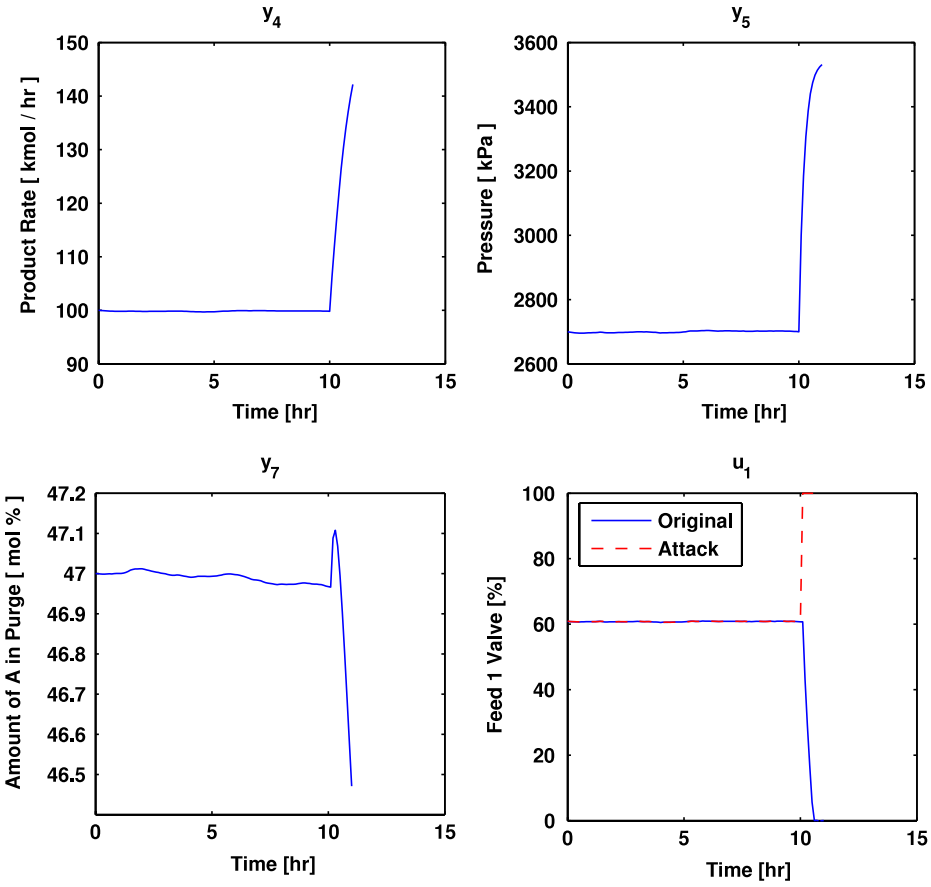

Fig. 9 – Integrity attack u_1^{max} from $t = 0$ to $t = 30$.

- Scaling Attacks:

$$\hat{y}_i^s(t) = \begin{cases} y_i(t) & \text{for } t \notin \mathcal{T}_a \\ \alpha_i(t) y_i(t) & \text{for } t \in \mathcal{T}_a \text{ and } \alpha_i y_i(t) \in \mathcal{Y}_i \\ y_i^{min} & \text{for } t \in \mathcal{T}_a \text{ and } \alpha_i y_i(t) < y_i^{min} \\ y_i^{max} & \text{for } t \in \mathcal{T}_a \text{ and } \alpha_i y_i(t) > y_i^{max}. \end{cases}$$

- Additive Attacks:

$$\hat{y}_i^a(t) = \begin{cases} y_i(t) & \text{for } t \notin \mathcal{T}_a \\ y_i(t) + \alpha_i(t) & \text{for } t \in \mathcal{T}_a \text{ and } y_i(t) + \alpha_i(t) \in \mathcal{Y}_i \\ y_i^{min} & \text{for } t \in \mathcal{T}_a \text{ and } y_i(t) + \alpha_i(t) < y_i^{min} \\ y_i^{max} & \text{for } t \in \mathcal{T}_a \text{ and } y_i(t) + \alpha_i(t) > y_i^{max}. \end{cases}$$

Similar attacks can be launched against controllers:

- Min and Max Attacks:

$$\hat{u}_i^{min}(t) = \begin{cases} u_i(t) & \text{for } t \notin \mathcal{T}_a \\ u_i^{min} & \text{for } t \in \mathcal{T}_a. \end{cases}$$

and

$$\hat{u}_i^{max}(t) = \begin{cases} u_i(t) & \text{for } t \notin \mathcal{T}_a \\ u_i^{max} & \text{for } t \in \mathcal{T}_a. \end{cases}$$

- Scaling Attacks:

$$\hat{u}_i^s(t) = \begin{cases} u_i(t) & \text{for } t \notin \mathcal{T}_a \\ \alpha_i(t) u_i(t) & \text{for } t \in \mathcal{T}_a \text{ and } \alpha_i u_i(t) \in \mathcal{U}_i \\ u_i^{min} & \text{for } t \in \mathcal{T}_a \text{ and } \alpha_i u_i(t) < u_i^{min} \\ u_i^{max} & \text{for } t \in \mathcal{T}_a \text{ and } \alpha_i u_i(t) > u_i^{max}. \end{cases}$$

- Additive Attacks:

$$\hat{u}_i^a(t) = \begin{cases} u_i(t) & \text{for } t \notin \mathcal{T}_a \\ u_i(t) + \alpha_i(t) & \text{for } t \in \mathcal{T}_a \text{ and } u_i(t) + \alpha_i(t) \in \mathcal{U}_i \\ u_i^{min} & \text{for } t \in \mathcal{T}_a \text{ and } u_i(t) + \alpha_i(t) < u_i^{min} \\ u_i^{max} & \text{for } t \in \mathcal{T}_a \text{ and } u_i(t) + \alpha_i(t) > u_i^{max}. \end{cases}$$

2.3. *Modeling DoS attacks*

In a DoS attack, we assume that a sensor signal does not reach the controller or that a control signal does not reach an actuator. Because the controller or actuator will notice the missing signal, it is necessary to implement functionality that enables the device to respond to this event.

Let $\hat{u}$ and $\hat{y}$ denote the response strategies for handling DoS attacks. A conservative response strategy uses the last signal received as the current command. In other words, the controller assumes that the missing sensor measurement is the same as the measurement it last received:

$$\hat{y}_i^{past}(t) = \begin{cases} y_i(t) & \text{for } t \notin \mathcal{T}_a \\ y_i(t_s) & \text{for } t \in \mathcal{T}_a. \end{cases}$$

A similar assumption can be made for a DoS attack on a control signal. In particular, we assume that an actuator continues operating based on the control signal corresponding to the manipulated variable value that it last received:

$$\hat{u}_i^{past}(t) = \begin{cases} u_i(t) & \text{for } t \notin \mathcal{T}_a \\ u_i(t_s) & \text{for } t \in \mathcal{T}_a. \end{cases}$$

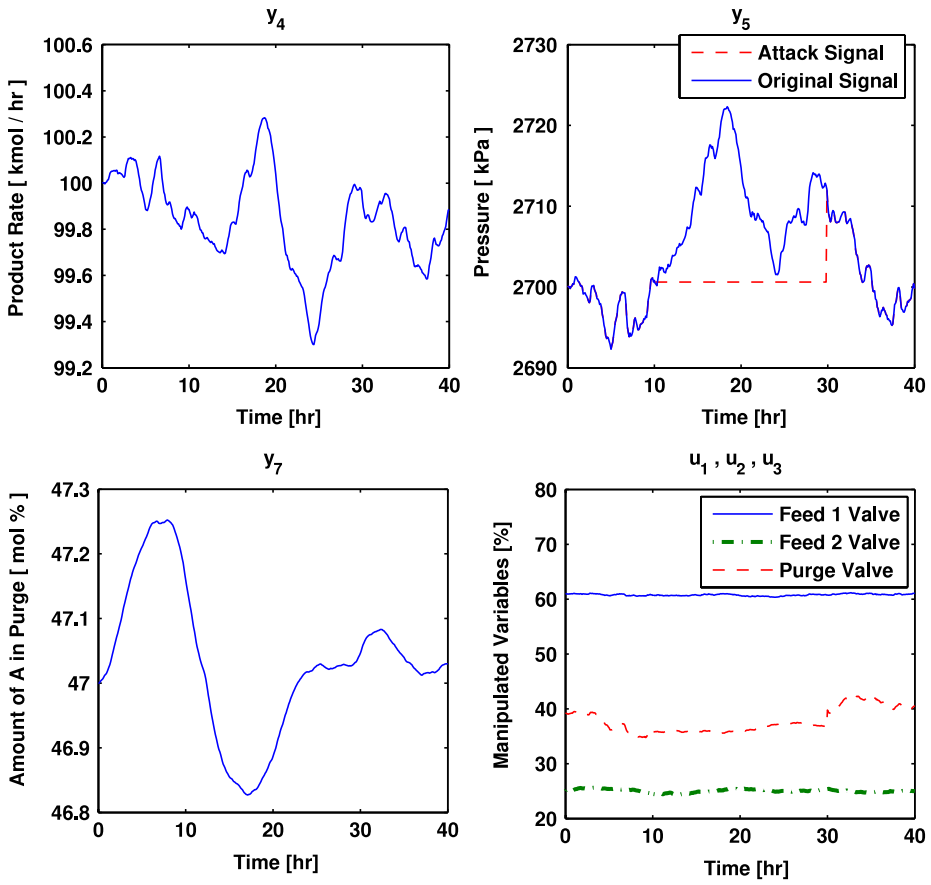

Fig. 10 – DoS attack on y_5.

3. Experimental results

This section describes the experimental setup and analyzes the experimental results.

3.1. *Chemical reactor system*

A chemical reactor system with a proportional integral (PI) control algorithm [12] is investigated in this paper. The dynamical model was coded in FORTRAN and the control algorithm in Matlab. The attacks were implemented using Matlab.

Fig. 3 shows the model of the chemical reactor system. Four chemical components are involved (*A*, *B*, *C*, and *D*). The goal of the control system is to maintain the irreversible reaction $A + C \xrightarrow{B} D$ at a specified rate while keeping the pressure inside the tank below 3000 kPa. Note that *B* is an inert component.

The chemical reactor system has three actuators. The first actuator, which is controlled by $u_1(t)$, operates a valve that controls feed F_1 containing the chemical components *A*, *B*, and *C*. The second actuator, controlled by $u_2(t)$, is a valve that controls feed F_2 containing *A*. The third actuator, controlled by $u_3(t)$, is a valve that purges the gas created by the chemical reaction. Each control signal $u_i(t)$ has a range between 0% (the valve is completely closed) and 100% (the valve is completely open).

The control algorithm [12] uses data from three sensors that monitor the product flow (y_4), pressure inside the tank (y_5), and amount of component *A* in the purge (y_7). Note that u_1 is a function of y_5 and y_4, u_2 is a function of y_7, and u_3 is a function of y_5.

Fig. 4 shows the chemical plant outputs without any noise inputs. Fig. 5 shows the plant outputs with Gaussian noise inputs. Specifically, Gaussian process noise (disturbance) with a mean of 0 and a variance of 0.05 is introduced at each valve. Note that the disturbances cause the system not to return to the steady state.

The chemical reactor system is simulated from $t = 0$ to $t = 40$ (h). Note that all the attacks in the experiments are executed from $t = 10$ to $t = 30$ (h).

3.2. *Integrity attacks*

We assume that the goal of the attacker is to raise the pressure inside the reactor vessel to an unsafe value (greater than 3000 kPa), causing equipment damage and possibly an explosion.

The integrity attacks (scaling, additive, and constant attacks) described in Section 2.2 were implemented. Only one sensor or controller was attacked at a time. The *max* and *min* attacks were the most effective; however, not all the attacks were able to drive the pressure to an unsafe level. We summarize the results below.

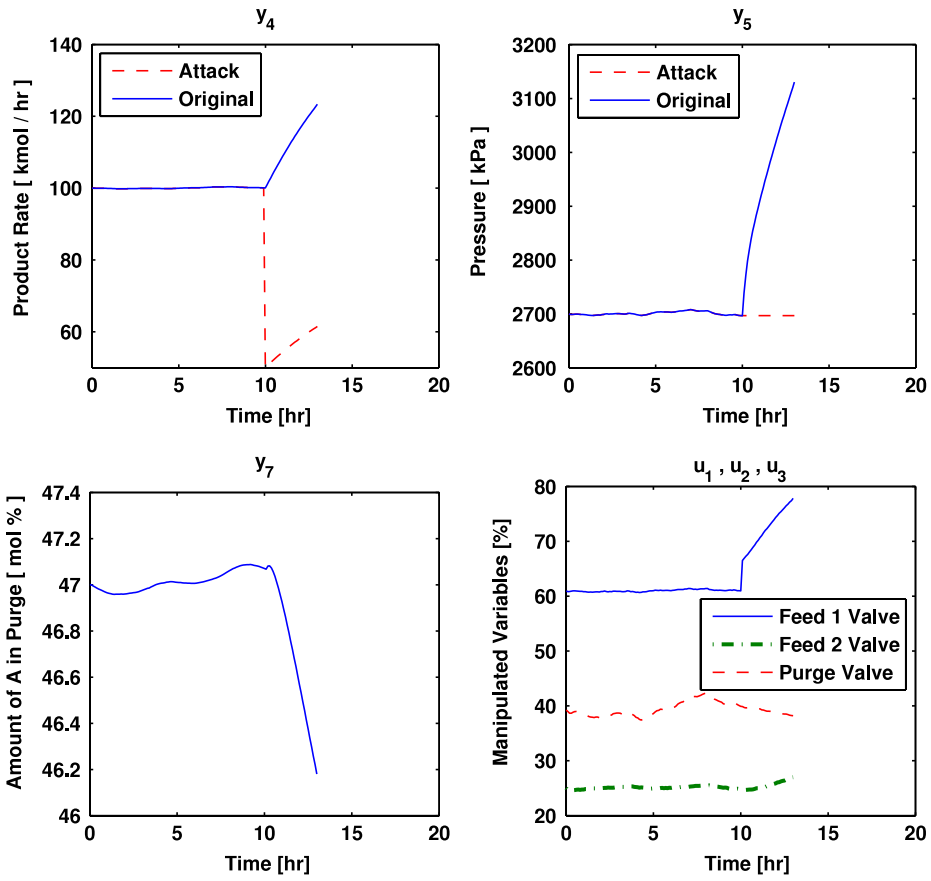

Fig. 11 – DoS attack on y_5 and integrity attack on y_4.

When a sensor is attacked, the controller can be expected to output an incorrect control signal because it operates on incorrect sensor information. If an attacker does not know the plant dynamics or the control algorithm, he/she may compromise a sensor at random. We assume the attacked sensor is y_7. Fig. 6 shows the effect of a y_7^{max} attack, which informs the controller that there is a large amount of component A in the reactor vessel. The simulations demonstrate that the plant returns to the steady (safe) state after the attack. Furthermore, the pressure in the reactor vessel is always below 3000 kPa.

Our experiments demonstrate that the chemical reactor system is very resilient to attacks on y_7, y_4, and u_2. Constant attacks are the most damaging, but they do not move the system to an unsafe state.

An attacker with knowledge about the system dynamics and control system operation would recognize that control signals u_1 and u_3 directly influence the pressure in the reactor vessel. Furthermore, the sensor that monitors the pressure in the reactor vessel tank y_5 would be an attractive target.

Fig. 7 shows the results of launching attack y_5^{min}. During the attack, the controller believes the pressure in the tank to be very low (0 kPa). Therefore, it shuts the purge valve with the goal of increasing the pressure. Because the sensor keeps sending the false pressure reading of 0 kPa, the controller keeps the purge valve shut for the duration of the attack. In our experiments, it took about 20 hours for the attack to increase the pressure above 3000 kPa (the unsafe state). This time period is long enough for plant operators to observe the unusual phenomenon and take the appropriate mitigation steps.

In the following, we discuss the effects of attacking control signals u_1 and u_3, which appear to be promising from an attacker's point of view.

Intuitively, it appears that shutting down the purge valve would increase the pressure. Therefore, we decided to launch attack $u_3^{min}(t)$. The results are shown in Fig. 8. The original signal computed by the controller is discarded and the attack forces the purge valve to close. This causes the chemical components to accumulate in the reactor vessel. However, although the accumulation raises the pressure from 2700 kPa to 2900 kPa (y_5 curve), it does not force the chemical reactor system to an unsafe state. The reason is that the control signal u_1 is also dependent on y_5; thus, when the pressure rises, the feed rate is correspondingly reduced.

Finally, we discuss the effects of launching attack $u_1^{max}(t)$ (Fig. 9). The original signal computed by the controller is discarded and the valve for Feed 1 is opened completely. In this case, large amounts of input flow to the reactor, causing the pressure to rise above 3000 kPa (y_5 curve). Note that this attack forces the system to an unsafe state in the shortest time.

We conclude that in order for a plant operator to prevent an attack from moving the system to an unsafe state, he/she should prioritize the protection of the control signal u_1. The sensor y_5 is also a priority. However, because elevating the

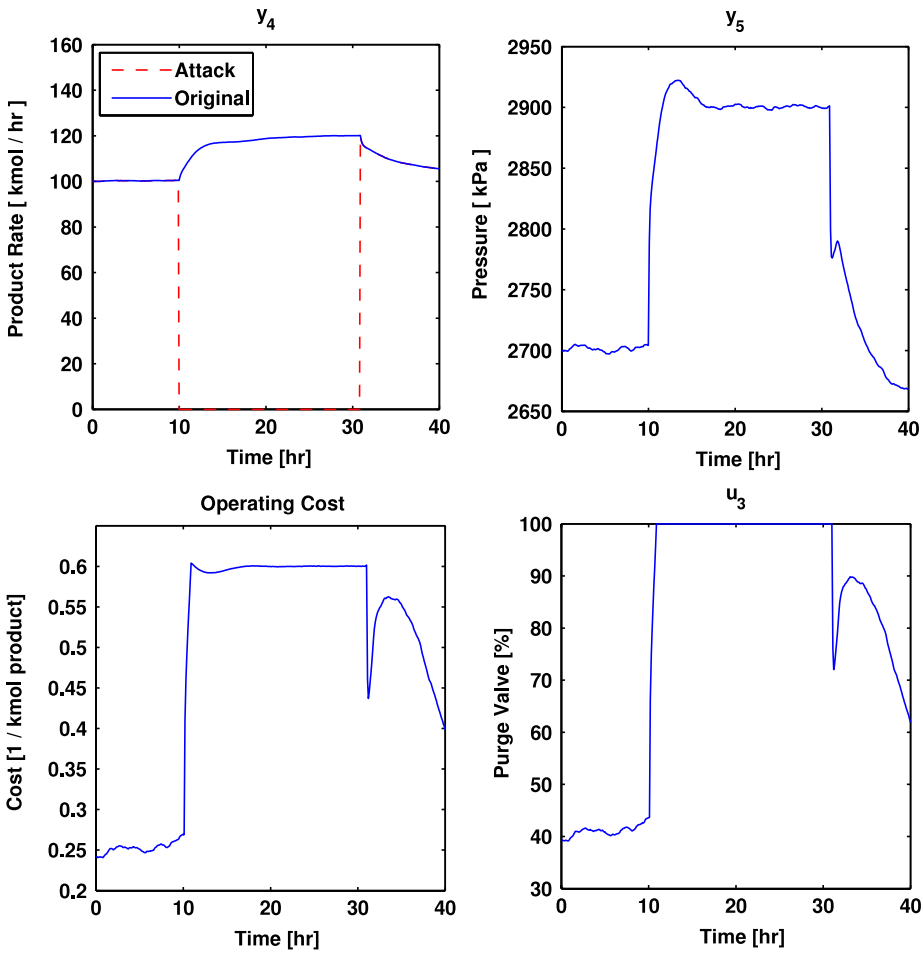

Fig. 12 – Integrity attack on the Loop 2 controller.

pressure by attacking y_5 takes a long time, the problem may be alleviated by monitoring the system and implementing the appropriate response when an anomaly is detected.

3.3. *DoS attacks*

Our experiments demonstrate that launching a DoS attack on a single device and implementing $\hat{u}^{past}$ or $\hat{y}^{past}$ does not have a major impact when the plant reaches a steady state. For example, note that the DoS attack on sensor y_5 in Fig. 10 does not cause the curve for y_5 to change significantly. Similar responses are obtained for all the other sensors and actuators. We conclude that the effects of DoS attacks on individual devices are limited and that protecting against integrity attacks should be a priority.

DoS attacks, however, can be launched in combination with innocuous integrity attacks to cause significant damage. Consider, for example, a DoS attack on y_5 coupled with an integrity attack on the production rate y_4 (which introduces a small variation of $y_4^s(t)$ with $\alpha = 0.5$). After the attacks are launched, the Loop 1 controller opens the Feed 1 valve to increase the production rate. This increases the flow of reactants to the reactor vessel, but the pressure sensor y_5, which is targeted by the DoS attack, fails to observe that the pressure in the vessel is rising. The resulting accumulation of reactants causes the pressure to exceed 3000 kPa in a fairly short time. Note that the changes to y_4 and y_5 in Fig. 11 start at time $t = 10$ when the attacks are launched.

3.4. *Operating cost attack*

Apart from forcing the chemical reactor system to an unsafe state, the attacker may wish to have a negative economic impact by increasing its operating cost. Such an attack is not easily detected and can produce large economic losses in the long term.

Estimating the cost of an attack in a typical information technology environment is often difficult because it is necessary to produce valuations for information loss (e.g., stolen data) and opportunity cost (e.g., DoS attack against an e-commerce website). However, estimating the cost of an attack on a control system is easier because the operating cost of a plant can be computed based on the reactants consumed and the production rate.

In our plant model, the instantaneous operating cost depends on the quantities of reactants A (y_{A3}) and C (y_{C3}) and Flow F_3 and Flow F_4. According to Ricker [12], the operating cost of the chemical plant is given by

$$cost = \frac{F_3}{F_4}(2.206y_{A3} + 6.177y_{C3}). \quad (1)$$

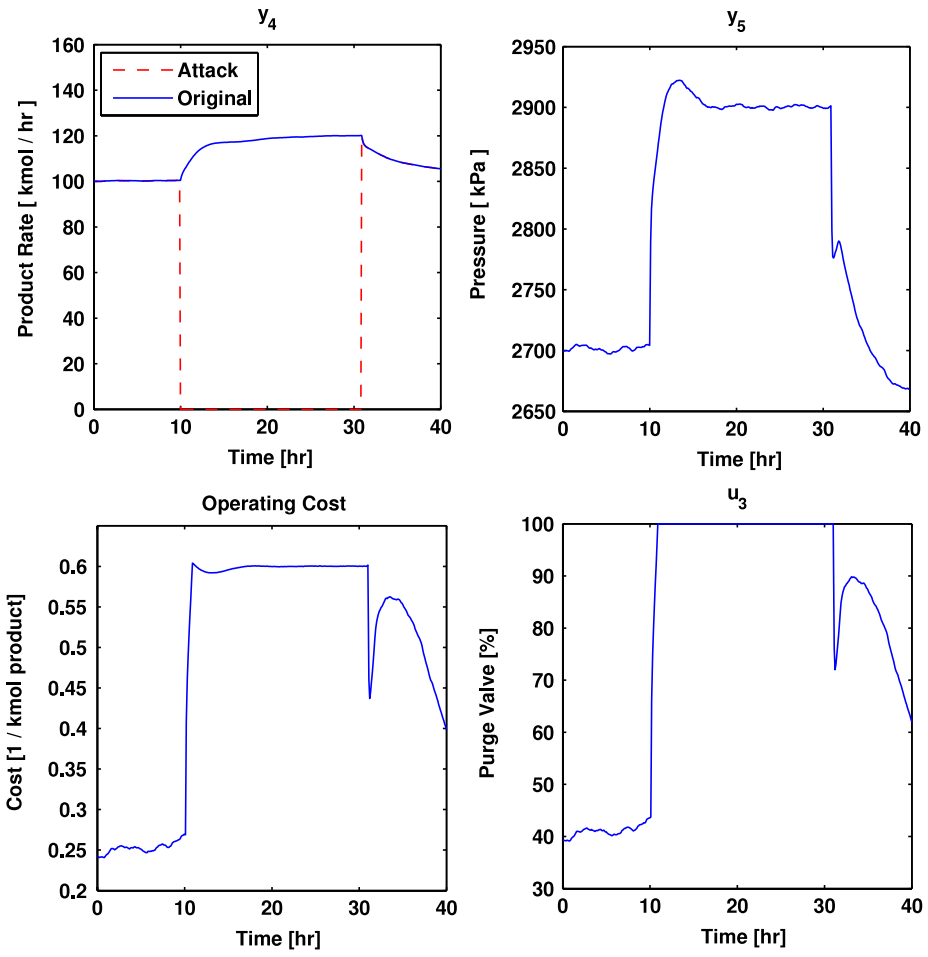

Fig. 13 – Integrity attack on y_4.

The operating cost is proportional to the purge flow (F_3) and the quantities of reactants A (y_{A3}) and C (y_{C3}) in the purge. Thus, an attacker may either target a controller to maximize the purge flow or target a sensor to confuse the controller and increase the quantities of the reactants A and C.

Now consider an attack on the Loop 2 controller. In this case, the purge valve is opened to increase the purge flow (larger F_3 value). Fig. 12 shows how the attack increases the operating cost of the plant (from $t = 10$ to $t = 30$).

Next, consider an integrity attack on sensor y_4 that sends an incorrect (zero) signal to the Loop 1 controller indicating that there is an insufficient quantity of reactants in the tank. In attempting to the maintain the production rate, the controller issues an incorrect control signal u_1 to increase the feed rate of A, B, and C by opening the Feed 1 valve. The increased quantity of reactants results in higher production flow (F_4) and higher reactor pressure (curve y_5 in Fig. 13). However, upon detecting the change in pressure, the Loop 2 controller turns on the purge valve to regulate the pressure. This increases the purge flow F_3, which leads to a higher operating cost, as shown in Fig. 13.

Based on the experiment results, we can conclude that targeting the purge flow valve is the most effective strategy for increasing the operating cost of the chemical reactor system.

4. Conclusions

Formal models of process systems, control systems, and attacks provide a powerful mechanism for reasoning about attacks and their consequences. The investigation of integrity and DoS attacks on a chemical reactor system reveals several important points. A DoS attack has relatively little impact on the system in steady state; however, a DoS attack launched in combination with an innocuous integrity attack can produce serious consequences. An attacker needs to identify and attack the key sensors in order to drive a system to an unsafe state; in the case of the chemical reactor, targeting the reactor pressure sensor is most effective as it rapidly causes the system to cross the safety threshold. In general, attacks on control signals are more serious than attacks on sensor signals. Finally, an attack on plant economy involves a radically different strategy that an attack on plant safety.

Our future research will attempt to develop systematic techniques for evaluating the impact of simultaneous attacks. Another area of focus is the design of automatic attack

detection and response mechanisms that can enhance the resilience of control systems.

Acknowledgements

We wish to thank Adrian Perrig, Bruno Sinopoli, Gabor Karsai, and Jon Wiley for useful discussions related to control systems security. This effort was partially supported by the International Collaboration for Advancing Security Technology (iCAST) and the Taiwan Information Security Center (TWISC) Projects under Grants NSC97-2745-P-001-001, NSC97-2918-I-009-005 and NSC98-2219-E-009-003, respectively.

REFERENCES

[1] E. Byres, Designing secure networks for process control, IEEE Industry Applications 6 (5) (2000) 33–39.
[2] E. Byres, J. Lowe, The myths and facts behind cyber security risks for industrial control systems, in: VDE Congress, 2004.
[3] E. Goetz, S. Shenoi (Eds.), Critical Infrastructure Protection, Springer, Boston, Massachusetts, 2007.
[4] V. Igure, S. Laughter, R. Williams, Security issues in SCADA networks, Computers and Security 25 (7) (2006) 498–506.
[5] T. Kilpatrick, J. Gonzalez, R. Chandia, M. Papa, S. Shenoi, Forensic analysis of SCADA systems and networks, International Journal of Security and Networks 3 (2) (2008) 95–102.
[6] P. Oman, E. Schweitzer, J. Roberts, Protecting the grid from cyber attack — Part 2: Safeguarding IEDs, substations and SCADA systems, Utility Automation & Engineering T&D 7 (1) (2002) 25–32.
[7] M. Papa, S. Shenoi (Eds.), Critical Infrastructure Protection II, Springer, Boston, Massachusetts, 2008.
[8] K. Stouffer, J. Falco, K. Kent, Guide to Supervisory Control and Data Acquisition (SCADA) and industrial control systems security – initial public draft, National Institute of Standards and Technology, Gaithersburg, Maryland, 2006.
[9] P. Tsang, S. Smith, YASIR: A low-latency, high-integrity security retrofit for legacy SCADA systems, in: Proceedings of the Twenty-Third IFIP TC 11 International Information Security Conference, 2008, pp. 445–459.
[10] United States Computer Emergency Readiness Team (US-CERT), Control Systems Security Program, U.S. Department of Homeland Security, Washington, DC. www.us-cert.gov/control_systems/index.html.
[11] A. Wright, J. Kinast, J. McCarty, Low-latency cryptographic protection for SCADA communications, in: Proceedings of the Second International Conference on Applied Security and Network Security, 2004, pp. 263–277.
[12] N. Ricker, Model predictive control of a continuous, nonlinear, two-phase reactor, Journal of Process Control 3 (2) (1993) 109–123.