Timing of Cyber-Physical Attacks on Process Control Systems

Marina Krotofil, Alvaro Cardenas, Kishore Angrishi

Chapter 3

TIMING OF CYBER-PHYSICAL ATTACKS ON PROCESS CONTROL SYSTEMS

Marina Krotofil, Alvaro Cardenas and Kishore Angrishi

Abstract This paper introduces a new problem formulation for assessing the vulnerabilities of process control systems. In particular, it considers an adversary who has compromised sensor signals and has to decide on the best time to launch an attack. The task of selecting the best time to attack is formulated as an optimal stopping problem that the adversary has to solve in real time. The theory underlying the best choice problem is used to identify an optimal stopping criterion, and a low-pass filter is subsequently used to identify when the time series of a process variable has reached the state desired by the attacker (i.e., its peak). The complexities associated with the problem are also discussed, along with directions for future research.

Keywords: Cyber-physical attacks, optimal stopping, secretary problem

1. Introduction

One of the growing research areas related to cyber-physical system security is developing threat models that consider an adversary who can manipulate sensor or actuator signals in order to drive a physical process to an undesired state. While many researchers have focused on the implications of manipulating signals, little work has attempted to understand the complexity and uncertainties associated with launching successful attacks and, in particular, finding the "best time" to launch an attack.

Attempting to disrupt a physical process without clearly understanding the consequences of the attack actions on the process is likely to result in a minor nuisance instead of an actual disruption – after all, breaking into a system is not the same a breaking a system.

This paper considers an attacker who can read a sensor signal for a given process variable and has to decide on a time to launch a denial-of-service (DoS) attack in order to "freeze" a certain process value above or below the setpoint stored in controller memory [5]. In doing so, the attacker deceives the controller

about the current state of the process and evokes compensating reactions that could bring the process into the state desired by the attacker (e.g., unsafe state). In order to achieve the attack goal faster, the attacker may opt to freeze one of the peak values of a process variable (low or high) to expedite process dynamics. Typical sensor signals in a process control environment fluctuate around the setpoint or track dynamic changes in the process. In both cases, the process variable exhibits a time series of low and high peaks. The attacker neither knows how high nor how low the process variable can span, nor which of the peak values should be chosen from among all the possible boundary states.

This paper formulates the challenge as an optimal stopping time problem for the attacker. In particular, it is formulated as a best choice problem (also known as the secretary problem), in which the adversary is presented with a time series of system states provided by sensor measurements and has to decide on the optimal time to attack. Because the best choice problem assumes non-correlated time measurements, it is necessary to discern upward or downward trends in process measurements (time correlations) and then identify when a local optimum has been reached. This is a non-trivial task in many real-world environments because sensor measurements can be noisy and can have sudden fluctuations.

2. Timing and Cyber-Physical Security

The miniaturization of processors has enabled them to replace analog components in many electronic products. The further integration of microprocessors with input and output system components has led to the evolution of microcontrollers. Microcontrollers are ubiquitous in applications ranging from consumer electronics to complex industrial systems. Microcontrollers are embedded in purpose-built computing systems used for myriad applications in the physical world. Collaborative environments comprising computational and communications elements that control physical entities with the help of sensors and actuators are called cyber-physical systems. Cyber abuses in the information technology domain do not generally depend on timing aspects. In certain instances, such as during race conditions, time-of-check to time-of-use vulnerabilities and cross-site scripting attacks that rely on gaining access to session cookies before they expire, the attacker has to ensure that the attack occurs within a tight window of time. In cyber-physical systems, however, timing is more critical because the physical state of a system changes continuously, and during the system evolution over time, some states might be more vulnerable to attacks than others. Timing plays an important role in cyber-physical systems because it characterizes the vulnerability of a system. For example, it may take minutes to observe a process change realized by an actuator action, hours to heat a tank of water or burn out a motor, and days to destroy centrifuges [6]. Understanding the timing parameters of a physical process enables an attacker to construct a successful attack as well as to maximize its impact (damage).

This paper focuses on industrial control systems, an aggregated term covering architectures, mechanisms and algorithms, that enable the processing of

physical substances and the manufacturing of end products. Over the past few decades, industrial plants have undergone tremendous modernization. Technology has become an enabler of efficiency as well as a source of problems. Panels of relays are now embedded computers and simple analog sensors are now IP-enabled smart transmitters [8] with multiple wired and wireless communications modes, numerous configuration modes and even web-servers, so that maintenance staff can calibrate and manage the devices from remote locations. Thus, the possibility of remote exploitation of industrial control systems and the physical processes they manage has become a reality.

3. Optimal Stopping Problem

The adversary's goal is to cause a tangible impact on the targeted process. In the physical domain, the attacker can either tamper with the sensor signals or modify the manipulated variables issued by the controller. This paper focuses exclusively on sensor signals. In particular, it is assumed that an attacker intends to drive the process to an unsafe state by deceiving a controller about the current state of the process and thus forcing it to take harmful compensating actions. To accomplish this, the attacker can force the controller to believe that a process variable is below or above its setpoint. One way to achieve this is to forge the process variable by means of an integrity attack that subverts a sensor-controller communications channel and manipulates messages.

If the sensor-controller communications channel is secured (e.g., using message authentication codes), then the attacker might opt to jam the channel to prevent the controller from receiving process measurement updates. This type of attack is referred to as a DoS attack on the sensor signal. As a rule, controllers store sensor signals in dedicated memory registers that are updated when a new value is received. During the DoS attack, the input register designed to store measurements from a particular sensor are overwritten by fresh values. Therefore, the last process value that reached the controller before the attack is used for system control over the duration of the attack. As a result, the controller would generate control commands based on the last measurement received. In a general sense, a DoS attack is similar to an integrity attack, the only difference being that the adversary does not wield direct influence on the "attack value." Instead, the adversary may take advantage of the timing parameters of an attack, such as the starting time t_a and the duration T_a.

In previous work [4, 5], we have shown that the impact of an industrial control system attack is sensitive to the specific state of the targeted system. In particular, an attack may only be effective if the process variable is above (or below) a certain threshold. The higher (or lower) the attack process variable is beyond the threshold, the greater the impact. Moreover, since a DoS attack is easy to detect, the attacker must achieve the disruption objective as soon as possible after the attack is launched. Therefore, the attacker should aim at launching a DoS attack at the time the process variable of interest reaches a more vulnerable state, i.e., a local maximum (or minimum).

The attacker faces the following problem: given a time series that exhibits a sequence of peaks and valleys of different amplitudes, select one of the peaks to launch a DoS attack in real time. If the attacker strikes too soon, the opportunity to have a greater impact on the system is lost (compared with if the attacker waits until the process variable reaches a higher (or lower) value). However, if the attacker waits too long, the process variable may not reach a more vulnerable state than previously observed and the attacker could miss the opportunity to cause maximal damage and even have the implanted attack tools (e.g., communications jammers and sensor malware) detected before the attack is launched.

The problem of selecting an opportune time to attack can be framed as an optimal stopping problem. This problem focuses on choosing the time to take a particular action based on sequentially-observed random variables in order to maximize an expected payoff. The optimal stopping decision task, in which the binary decision to stop or continue the search depends only on the relative ranks, is modeled as the best choice problem, which is also known as the secretary problem [2].

3.1 Secretary Problem

In the standard version of the secretary problem, a finite and known number of items (or alternatives) n are presented to a decision maker sequentially and one-at-a-time in random order. Time is assumed to be discrete. At any period, the decision maker can rank all the items that have been observed in terms of their desirability or quality. For each item inspected, the decision maker must either accept the item, in which case the search process is terminated (reject), the next item in the random order is presented and the decision maker faces the same problem as before. The decision maker's objective is to maximize the probability of selecting the best item from among the n items available.

The classical secretary problem, which seeks to choose the best secretary from among all the applicants, has six assumptions:

- There is only one position available.
- The number of applicants n is finite and known to the decision maker.
- The n applicants are interviewed sequentially, one-at-a-time and in random order. Consequently, each of the $n!$ orders is equally likely.
- The decision maker can rank all n applicants from best to worst without ties. The decision to accept or reject an applicant in a given period is based only on the relative ranks of the applicants interviewed to that point.
- An applicant who is rejected cannot be recalled later.
- The decision maker is satisfied with nothing but the best. The payoff is one if the best applicant of the n applicants is selected; otherwise, the payoff is zero.

Note that an applicant is accepted only if the applicant is relatively the best among the applicants who have already been observed. A relatively best applicant is called a candidate.

The optimal stopping rule suggests that the best candidate can be selected with maximum probability $1/e$ using the rule: do not make an offer to the first n/e candidates and after that make an offer to the first candidate whose value exceeds the values of all the candidates seen thus far (or proceed to the last applicant if this never occurs). In other words, the algorithm starts with a learning phase in which the decision maker sees n/e candidates and sets an aspiration level equal to the highest value seen during the learning phase. After that, the decision maker hires the first candidate who exceeds the aspiration level.

The secretary problem assumptions impose more constraints on observation and selection than generally apply in practice [3]. Relaxing one or more assumptions to produce a more realistic formulation of the standard secretary problem has attracted the attention of the research community. This paper considers the classical solution along with a recent result that assumes the order in which the candidates arrive is not completely random, but has a probability distribution satisfying a hazard rate condition [7]. This assumption is commonly used in engineering applications – specifically, given that the value of a candidate is not less than y, the likelihood that it is equal to y increases as y increases. Gaussian, uniform and exponential distributions satisfy this property. Under these assumptions, it has been shown that the learning period falls from n/e to $n/log(n)$, meaning that it is enough to observe a much smaller number of candidates to set the optimal aspiration level. In a process control environment, the probability of detecting an intrusion increases with time, therefore, having a shorter learning phase is beneficial to the attacker.

3.2 Dealing with Correlated Time Series

While the secretary problem matches the problem that an attacker faces in our scenario, an additional condition that an attacker of a physical process encounters is that sensor signal samples do not arrive in random order. Instead, their time series represent continuous real-time measurements of physical phenomena and each sample X_i is heavily correlated with the next sample X_{i+1}. Thus, if a process variable (e.g., temperature) is increasing, it cannot drop radically in the next time instance.

Recall that the attacker sets the aspiration level to a value equal to the highest sample seen during the learning phase. According to the optimal solution algorithm for the secretary problem, upon completing the learning phase the attacker should select the first sample whose value exceeds the aspiration level. By doing so, the attacker would miss the opportunity to select an even higher value as in the case of an upward trend, where the process measurements keep increasing until a local peak is reached. Hence, unlike the static choice rule discussed above, the attacker may incorporate expectations about the future in the decision process. In this case, the choice between stopping and continuing

to search at sample X_i is determined not only by the aspiration value but also by the difference between the stopping value and the continuation value X_{i+1}. The problem of identifying a signal peak is exacerbated by the fact that process variables are noisy and, therefore, an upward trend might be followed by a quick drop, followed again by an even higher gain.

To solve this problem, a low-pass filter is incorporated to smooth out short-term signal fluctuations and highlight the longer-term trends. This enables a peak to be identified as soon as a downward trend in a smoothed signal is detected (e.g., three consecutive measurement drops).

4. Simulation Setup

The empirical analysis employed a Matlab model of the Tennessee Eastman challenge process [1] developed by Ricker [9]. It is implemented as a C-based MEX S-function with a Simulink model.

4.1 Tennessee Eastman Challenge Process

The Tennessee Eastman challenge process [1] is a modified model of a real plant-wide industrial process. The process produces two liquid (l) products from four gaseous (g) reactants involving two irreversible exothermic reactions:

$$\begin{array}{ll} A(g) + C(g) + D(g) \rightarrow G(l) & \text{Product 1} \\ A(g) + C(g) + E(g) \rightarrow H(l) & \text{Product 2.} \end{array}$$

Figure 1 shows the Tennessee Eastman challenge process. It incorporates five major units: reactor, condenser, vapor-liquid separator, recycle compressor and stripper. The gaseous reactant and products are not specifically identified. Feed C is not pure and consists of 48.5% A and 51% C. The gas phase reactions are catalyzed by a substance dissolved in the liquid phase in the reactor. The products and unreacted ingredients leave the reactor in the vapor phase, pass through the condenser and then proceed to the vapor-liquid separator. Non-condensed components cycle back to the reactor via the recycle compressor. Condensed components are sent to the stripper that removes the remaining reactants. The byproducts and inerts are purged from the system in the vapor phase using the vapor-liquid separator whereas products G and H exit the stripper base and are separated in the downstream refining section.

The plant has eleven valves for manipulation and 41 measurements for process monitoring. In the simulation model, the control configuration involves eighteen proportional-integral (PI) controllers, sixteen process measurements XMEAS{1; 2; 3; 4; 5; 7; 8; 9; 10; 11; 12; 14; 15; 17; 31; 40} and nine setpoints that form eight multivariable control loops and one single feedback control loop [5]. All the process measurements include Gaussian noise with standard deviations typical of the types of measurements. The default simulation time for a single experiment is 72 hours with a sampling frequency of 100 measurement samples per hour. Timestamps of the simulated data sets are stored in the designated variable `tout`.

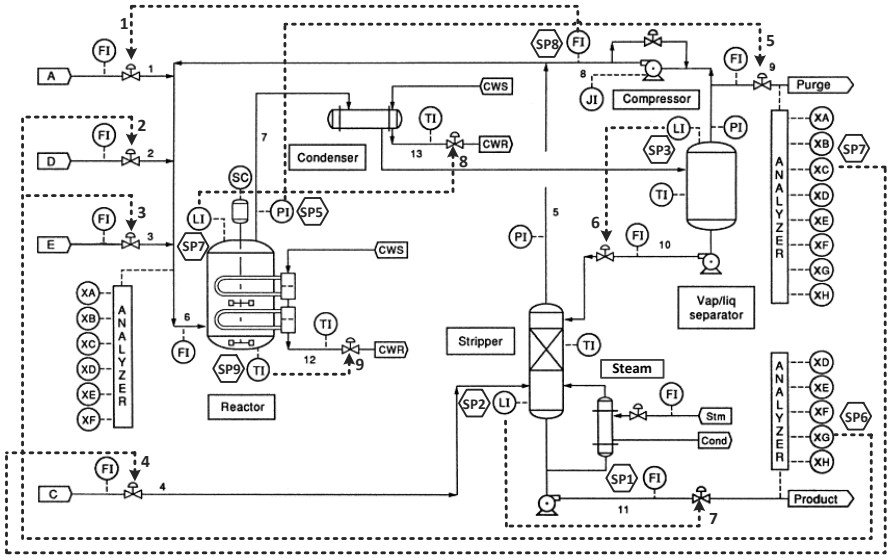


Figure 1. Tennessee Eastman challenge process [9].

In order to obtain statistically significant results, the original code was modified by generating a new seed for the random number generator for each run. In addition, higher sampling rates for the process variables – 2,000 sensor samples per hour (per sensor) – were incorporated in the Matlab workspace.

4.2 DoS Attack Modeling

Let $X_i(t)$ be the measurement by sensor i at time t where $0 \leq t \leq T$ and T be the duration of the simulation. The attack interval T_a is arbitrary and is limited to the simulation run time. The manipulated sensor readings X_i' are simulated as follows:

$$X_i'(t) = \begin{cases} X_i(t), & \text{for } t \notin T_a \\ X_i^a(t), & \text{for } t \in T_a \end{cases}$$

where $X_i^a(t)$ is the modified reading (attack value).

During a DoS attack, sensor signals do not reach the controller. If the attack starts at time t_a, we have:

$$X_i^a(t) = X_i(t_a - 1).$$

This is translated to the attacker's goal as follows: as soon as the peak is identified and the process value starts decreasing again, the attacker should

immediately launch a DoS attack to freeze the peak value from the previous control loop cycle in the controller memory.

4.3 Low-Pass Filter for Sensor Signals

The simplest form of signal smoothing is the moving average, which corresponds to the mean of the previous N data points. If μ is the smoothing interval, then the moving mean is given by:

$$\hat{x}_n = \begin{cases} \hat{x}_{n-1} - \frac{x_{\mu-n}}{n} + \frac{x_\mu}{n} & \text{for } n > \mu \\ \frac{n-1}{n} \cdot \hat{x}_{n-1} + \frac{x_n}{n} & \text{for } n < \mu. \end{cases}$$

One of the side-effects of signal smoothing is the delay of the smoothed signal with respect to the original signal by $(\mu - 1)/2$ samples. To avoid shifting data in financial applications, it is recommended to average the same number of values before and after the average is calculated. However, this is not possible during real-time analysis. As a result, when the smoothed signal reaches its peak, the real measurement is already decaying. Another factor to consider is signal amplitude reduction. Increasing the smoothed signal width improves the signal-to-noise ratio but reduces the peak height. Because the aspiration value is determined based on the smoothed signal, it is not optimal.

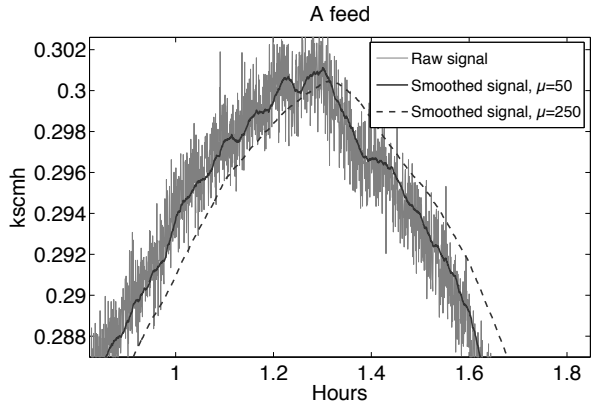
Figure 2 shows the smoothing results for a sensor signal smoothed over different smoothing intervals. As can be seen, when μ is too small, smoothing does not sufficiently remove the noise (Figure 2(a)). As a result, stopping decisions are taken before the state reaches its local peak (Figure 2(b)).

To mitigate this problem, we introduce a retry parameter r. If $r = 0$, the search stops if the current sample is smaller than the previous sample because this could indicate that the peak has been determined and the process value is falling. Correspondingly, if $r = 3$, the search is stopped if three consecutive samples are smaller than the last "peak" sample. As discussed in the next section, the retry parameter plays an important role in the success of an attack.

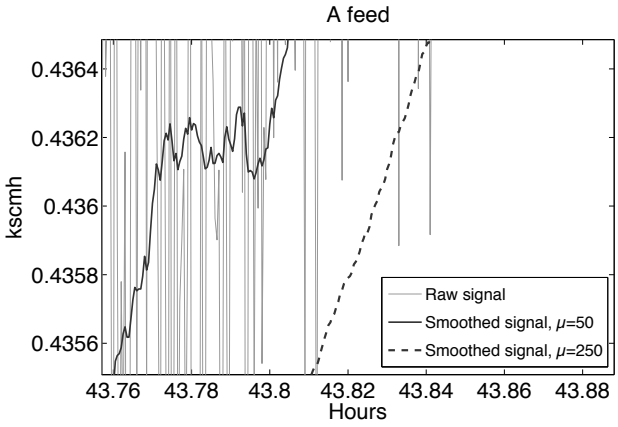
In the Tennessee Eastman process, sensor signals can be roughly divided into four groups (Figure 3). A Type 1 sensor signal has large variations with low noise levels (XMEAS{1; 10; 11}). A Type 2 signal measures a variable that is at steady-state but has high frequency noise (XMEAS{2; 3; 9; 17}). A Type 3 signal is a noisy variation of a Type 1 signal (XMEAS{4; 5; 8; 12; 14; 15}). A Type 4 signal has multiple noisy signal peaks (XMEAS7). The next section shows that, in order for an attacker to successfully conduct an attack, it is necessary to consider the type of signal that will be exploited.

5. Experimental Results

The experiments assume the presence of an attacker whose goal is to force the physical process to shut down. The result of such an attack is evaluated using the shutdown time (SDT), the time that the process is able to run before being shut down because it has exceeded the safety constraints. First, the



(a) Smoothing effect.



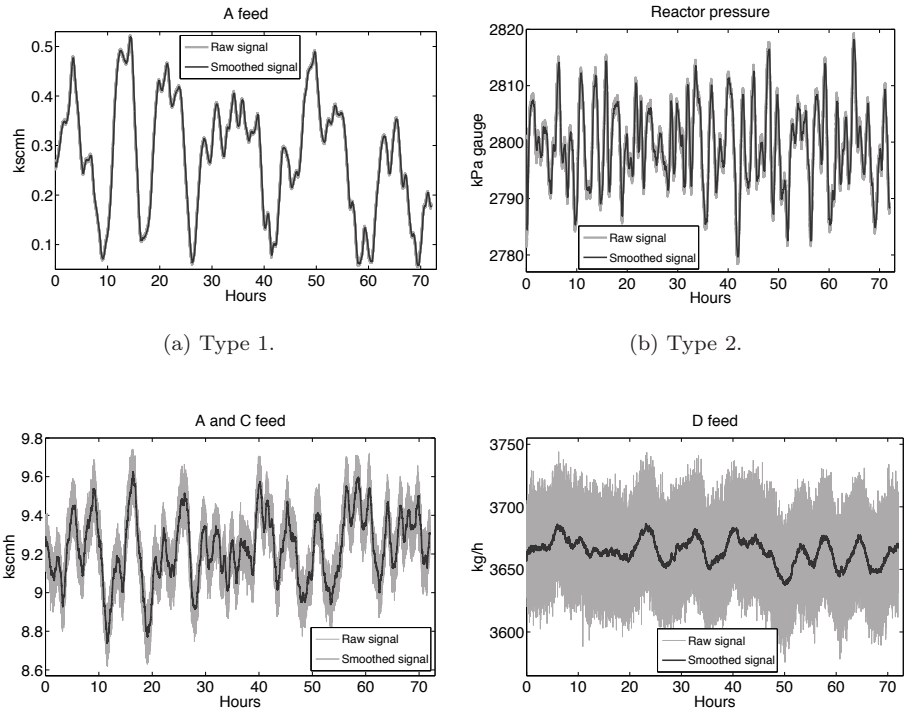
(b) Smoothing artifacts.

Figure 2. Signal smoothing.

shortest SDT that can be achieved using a DoS attack on each sensor signal is determined. Following this, to justify the importance of the strategic selection of the attack time, evidence of the ineffectiveness of DoS attacks conducted at random times is provided. In particular, it is shown that random selection not only significantly increases the time required to bring the process to the critical state, but in some cases, it could be completely ineffective. Also, the experiments evaluate the effects of the length of the learning phase and parameter smoothing on the attacker's prospects of selecting the highest (or lowest) possible process value in real time.

5.1 Shortest Shutdown Time

To find a reference value for the worst-case attacks, the lowest and highest possible process values based on the results of 20 simulations were determined.



(a) Type 1.

(b) Type 2.

(c) Type 3.

(d) Type 4.

Figure 3. Different sensor signals and their smoothed versions (μ=250).

These can be considered to be the optimal attacks (but practically infeasible because the attacker has to analyze the signals and launch the attacks in real time). As $X_i^a(t)$, we use:

$$X_i^{min}(t) = \min_{t \in T} X_i(t) \quad \text{and} \quad X_i^{max}(t) = \max_{t \in T} X_i(t).$$

The mean times to shutdown for the attacks on different sensors were determined based on the results of 50 simulations. Table 1 summarizes the results. The 95% confidence intervals are calculated using the Student's t-distribution. The table does not include results for XMEAS{10; 11} because no attack on these sensors drives the system to an unsafe state.

Due to the variability of process measurement noise, the process is never in the same state. However, as the results indicate, the Tennessee Eastman process is, in general, resilient to noise variations and the SDT does not exhibit significant variations, with the exception of the attacks $F_{recycle}^{max}$ and F_A^{min}.

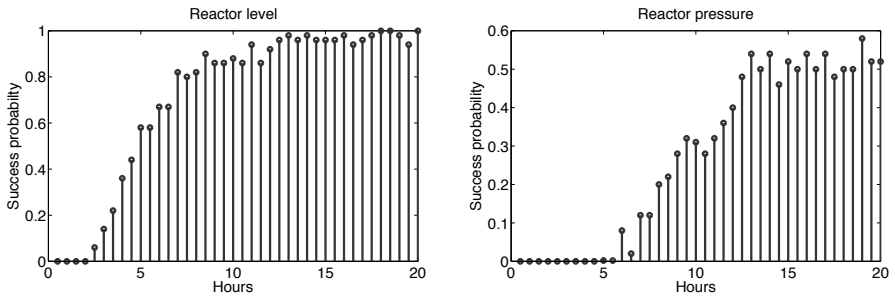
Table 1. Simulation results for the process-aware attack strategy.

XMEAS	Variable	Units	Min/ Max	SDT (h)	Confidence Interval (95%)
(1)	A-Feed Rate	kscmh	0.0487/ 0.7466	12.116 –	(4.919; 19.310) –
(2)	D-Feed Rate	kg h^{-1}	3,556/ 3,750	3.840 3.489	(3.641; 4.040) (3.387; 3.590)
(3)	E-Feed Rate	kg h^{-1}	4,322/ 4,553	4.120 2.672	(3.916; 4.427) (2.517; 2.879)
(4)	C-Feed Rate	kscmh	8.524/ 9.825	0.284 0.920	(0.263; 0.305) (0.826; 1.026)
(5)	Recycle Flow	kscmh	29.32/ 35.17	3.824 7.324	(3.384; 4.153) (6.358; 8.773)
(7)	Reactor Pressure	kPa	2,771/ 2,829	8.300 –	(7.811; 8.638) –
(8)	Reactor Level	%	60.73/ 68.27	1.877 2.363	(1.778; 1.976) (2.100; 2.482)
(9)	Reactor Temperature	°C	122.86/ 123	1.310 0.374	(1.265; 1.346) (0.370; 0.381)
(12)	Separator Level	%	38.49/ 61.2	4.913 3.277	(4.726; 5.184) (3.168; 3.397)
(14)	Separator Underflow	m^3 h^{-1}	24.12/ 26.87	7.241 5.584	(6.847; 7.672) (5.168; 5.930)
(15)	Stripper Level	%	29.17/ 72.96	5.189 4.990	(4.900; 5.375) (4.880; 5.120)
(17)	Stripper Underflow	m^3 h^{-1}	22.37/ 23.5	1.287 0.932	(1.020; 1.634) (0.910; 0.960)

Attack F_A^{min} on the *A*-feed is of special interest. Not all attack instances trigger process shutdowns. Thus, the result for the F_A^{min} attack is based on 43 out of 50 cases where the process reaches an unsafe state. At the same time, attacks $P_{pressure}^{max}$ and F_A^{max} do not drive the process to an unsafe state. This means that an attacker who intends to launch an attack on reactor pressure should only strike at the minimum peaks.

5.2 Random Attack Strategy

The outcome of a DoS attack at a random time results in an arbitrary value being stored in controller memory. The closer the attack value to the setpoint, the more time it takes for the process to reach an unsafe state. To evaluate



(a) XMEAS8. (b) XMEAS7.

Figure 4. Probability of success.

the effectiveness of launching a DoS attack at a random time, we compute the probability of the process reaching the safety limits based on 100 simulations for different DoS attack durations T_a.

The previous section noted that both the process-aware attacks $L^{max}_{reactor}$ and $L^{max}_{reactor}$ on XMEAS8 take about two hours to bring the process to an unsafe state. For purposes of comparison, Figure 4(a) shows the time taken to move the process to an unsafe state by striking randomly. Note that the attack would have to continue for at least seven hours to achieve reliable results (e.g., 75% probability). Furthermore, Figure 4(b) shows that, without process knowledge, the attacker cannot reliably succeed in launching an attack on XMEAS7.

Notably, it is almost impossible to execute a successful attack on XMEAS1 by conducting a random DoS attack. This is because the susceptibility of the process to an attack on the A-feed depends greatly on the attack value as well as the overall system state. Because a fresh stream of C contains 48.5% of A, the control scheme carefully maintains a stoichiometric balance of A and C in the system. As a result, certain attacks on XMEAS1 would be compensated for by the system.

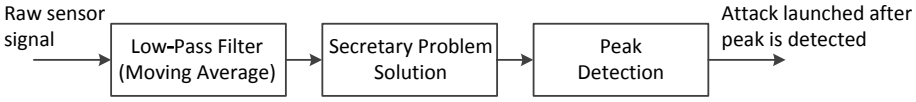


Figure 5. Generalized approach.

5.3 Optimal Stopping Attack Strategy

The results in the preceding section demonstrate that the adversary cannot achieve the attack goal fast and/or reliably enough without strategic decision making with respect to the attack time. This section analyzes the attacker's prospects of selecting the highest possible process value in real time by applying the strategies described in the previous sections (Figure 5).

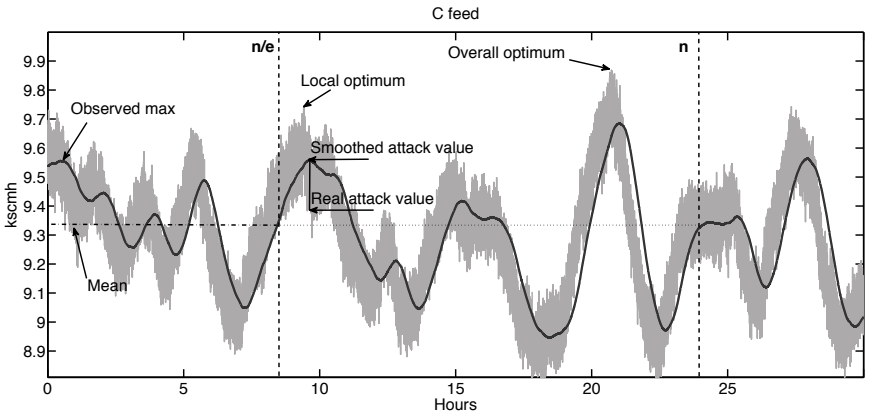


Figure 6. Educated guess approach.

Figure 6 shows the implementation of our approach. To begin, the attacker has to decide on the two parameters of the secretary problem, namely the number of samples or alternatives to consider (n) and the duration of the learning phase. For simplicity, n is measured in hours. For a time frame of 24 hours, the number of alternatives is equal to $24 \times f_s$ where f_s is the sampling rate of the sensor signal ($f_s = 2,000$). Next, the attacker must choose the smoothing parameter μ and retry parameter r. Experiments were conducted to decide on the appropriate smoothing interval; they yielded three values for the analysis: $\mu = \{50; 150; 250\}$. Similarly, reasonable values for the retry parameter were found to be in range $r = \{0; 1; 2; 3\}$.

The attacker begins the smoothing of the signal and conducts the selection process in real time. The aspiration level (reference value) is set based on the greatest value of the smoothed signal observed during the learning period. Upon completing the learning phase, the attacker sequentially inspects every sample of the smoothed signal until a sample is found whose value exceeds the reference value. Following this, the attacker applies the forward-looking strategy described in Section 3.2. Next, the value of the sample $\hat{x}_i$ is checked to see if it exceeds the previous one $\hat{x}_{i-1}$. If not, the search continues until the condition $\hat{x}_i > \hat{x}_{i-1}$ is met, because this may indicate that the process measurement has reached its peak and has started to decay. The value of the retry parameter determines how many times the latter condition should be met before making the final stopping decision. In this case, the real attack value is equal to the value of the raw signal sample X_i^a at time $(t_a - 1)$.

Next, we evaluate the performance of the approach based on three metrics: (i) fractional error in identifying the peak (as a percentage) to measure the effectiveness of the low-pass filter and retry parameter r; (ii) fractional error in selecting the highest possible value in the time series (as a percentage) to measure the effectiveness of the stopping problem solution; and (iii) number of non-selections (last sample in the series is taken) evaluated as the average fractional error in selecting the largest possible sample.

Table 2 summarizes the results of applying the strategies to XMEAS1 (Type 1 signal) based on 50 simulations. The simulation results confirm that the learning period can be indeed cut down to $n/log(n)$ while producing results comparable with the n/e strategy. Due to the short learning period, the number of non-selections is reduced substantially (almost to zero). For the same reason, the fractional error in selecting the highest possible process value increases because the attacker has less time to achieve a sufficient aspiration level. Since the classic secretary problem solution results in an average of 25% non-selections, it can be a decisive factor to favor the $n/log(n)$ strategy.

The results also indicate that the appropriate selection of the smoothing factor significantly reduces the fractional error in selecting the highest possible alternative. Meanwhile, the retry parameter has a similar influence on the reduction of the fractional error in identifying the peak. The conclusion from the simulation results is that when planning an attack on a sensor signal of Type 1, the attacker should opt for the attack parameters $\mu = 250$ and $r = \{1; 2\}$ with learning window $n/log(n)$.

Finally, we demonstrate the performance for different types of sensor signals using histograms of the fractional errors in selecting the highest possible values in the corresponding time series (Figure 7). Note that the best results are obtained for sensor signals of Types 1 and 4. In contrast, the methodology proposed in this paper is not well suited to conducting attacks on sensor signals of Types 2 and 3 because of their noise levels. While applying a low-pass filter yields good results for attacks on low-noise signals, an alternative approach is required for dealing with noisy process variables. One possible approach, which we will examine in our future research, involves the use of non-parametric change detection statistics.

6. Conclusions

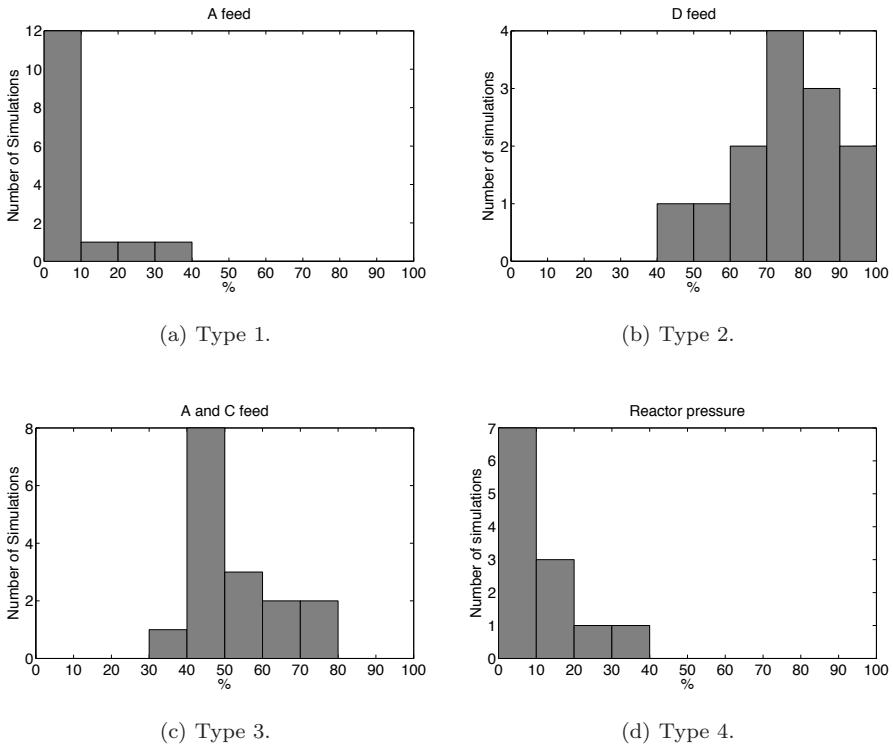
This paper demonstrates that sensor signal characteristics must be considered carefully when developing attacks that target process measurements. Moreover, finding the appropriate values of parameters such as optimal signal smoothing (μ) and stopping decision (r) are not straightforward and the parameters are best determined experimentally.

An attacker may do extensive homework and proactively design portions of attacks, but the attacks would have to be tuned through reconnaissance activities such as changing configuration parameters, manipulating process variables and turning components on and off while observing the effects on the process system. From the defensive perspective, short-term process deviations arising from such "testing" can be detected by process-aware anomaly detection methods. Furthermore, in order to hinder the attacker's ability to disrupt a process system, plant administrators should strategically place misleading or false technical documentation to influence the attacker's strategy selection.

Overall, a better understanding of the complexities and uncertainties faced by an attacker when designing targeted cyber-physical attacks in the physical domain allows for better judgment regarding the efforts required to design

Table 2. Simulation results for the educated guess approach for XMEAS1.

	$\mu = \mathbf{50}$			
	$\mathbf{r = 0}$	$\mathbf{r = 1}$	$\mathbf{r = 2}$	$\mathbf{r = 3}$
$\frac{\mathbf{N}}{\mathbf{e}}$	• 1.06 (0.82; 1.32) • 49.46 (31.8; 67.11) • 7 (79.07)	• 0.7 (0.45; 0.96) • 44.86 (29.91; 59.81) • 4 (96.17)	• 0.62 (0.25; 0.98) • 35.78 (20.36; 51.20) • 6 (98.46)	• 0.82 (0.53; 1.11) • 54.08 (36.67; 73.04) • 8 (47.87)
24h	$\mu = \mathbf{150}$			
	• 1.35 (1.03; 1.67) • 33.98 (11.14; 56.83) • 8 (112.56)	• 0.74 (0.48; 1.00) • 26.56 (8.06; 45.50) • 5 (61.35)	• 0.84 (0.55; 1.14) • 22.47 (9.73; 35.20) • 5 (35.39)	• 0.75 (0.35; 1.16) • 21.39 (7.01; 35.76) • 5 (89.16)
	$\mu = \mathbf{250}$			
	• 1.42 (0.78; 2.07) • 26.87 (7.59; 46.04) • 7 (60.87)	• 0.73 (0.18; 1.29) • 8.56 (0.34; 16.78) • 3 (93.19)	• 0.80 (0.22; 1.38) • 6.33 (0.18; 12.49) • 7 (90.38)	• 0.69 (0.40; 0.98) • 27.96 (6.88; 49.03) • 5 (32.93)
$\frac{\mathbf{N}}{\mathbf{log(N)}}$	$\mu = \mathbf{50}$			
	• 1.31 (0.98; 1.69) • 72.63 (57.62; 87.66) • 0	• 0.75 (0.46; 1.04) • 65.48 (51.79; 79.16) • 0	• 0.84 (0.58; 1.09) • 66.16 (50.76; 81.55) • 0	• 0.69 (0.47; 0.84) • 74.56 (59.88; 89.23) • 1 (91.32)
24h	$\mu = \mathbf{150}$			
	• 1.04 (1.21; 1.59) • 49.73 (32.81; 66.65) • 0	• 1.03 (0.62; 1.45) • 48.26 (34.72; 61.35) • 1 (164.46)	• 1.00 (0.52; 1.47) • 33.6 (18.77; 48.42) • 0	• 0.96 (0.51; 1.43) • 27.85 (12.22; 43.49) • 0
	$\mu = \mathbf{250}$			
	• 1.49 (1.01; 1.98) • 37.57 (20.98; 54.7) • 0	• 0.71 (0.27; 1.15) • 40.44 (23.00; 57.86) • 1 (53.81)	• 0.78 (0.43; 1.12) • 28.33 (13.42; 43.25) • 0	• 0.84 (0.52; 1.16) • 46.74 (32.18; 61.29) • 0



(a) Type 1.

(b) Type 2.

(c) Type 3.

(d) Type 4.

Figure 7. Distributions of fractional errors for sensor signals ($\mu = 250$, $r = 2$).

and conduct cyber-physical attacks with surgical precision (as in the case of Stuxnet). Clearly, developing sophisticated and effective cyber-physical attacks requires extensive experimentation with the same specialized industrial equipment as that installed at the targeted site.

References

[1] J. Downs and E. Vogel, A plant-wide industrial process control problem, *Computers and Chemical Engineering*, vol. 17(3), pp. 245–255, 1993.

[2] P. Freeman, The secretary problem and its extensions: A review, *Revue Internationale de Statistique*, vol. 51(2), pp. 189–206, 1983.

[3] J. Gilbert and F. Mosteller, Recognizing the maximum of a sequence, *Journal of the American Statistical Association*, vol. 61(313), pp. 35–73, 1966.

[4] Y. Huang, A. Cardenas, S. Amin, Z. Lin, H. Tsai and S. Sastry, Understanding the physical and economic consequences of attacks on control systems, *International Journal of Critical Infrastructure Protection*, vol. 2(3), pp. 73–83, 2009.

[5] M. Krotofil and A. Cardenas, Resilience of process control systems to cyber-physical attacks, *Proceedings of the Eighteenth Nordic Conference on Secure IT Systems*, pp. 166–182, 2013.

[6] R. Langner, To Kill a Centrifuge: A Technical Analysis of What Stuxnet's Creators Tried to Achieve, The Langner Group, Arlington, Virginia, 2013.

[7] M. Mahdian, R. McAfee and D. Pennock, The secretary problem with a hazard rate condition, *Proceedings of the Fourth International Workshop on Internet and Network Economics*, pp. 708–715, 2008.

[8] C. McIntyre, Using smart instrumentation, *Control Engineering* (`www.controleng.com/single-article/using-smart-instrumentation/a0ec350155bb86c8f65377ba66e59df8.html`), April 8, 2011.

[9] N. Ricker, Tennessee Eastman Challenge Archive, Department of Chemical Engineering, University of Washington, Seattle, Washington (`depts.washington.edu/control/LARRY/TE/download.html`), 2014.