Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2016, Article ID 4147251, 12 pages
http://dx.doi.org/10.1155/2016/4147251

Research Article

Vulnerability Analysis of CSP Based on Stochastic Game Theory

Jiajun Shen[1,2,3] and Dongqin Feng[1,2,3]

[1]*National Engineering Laboratory of Safety & Security Technology of Industrial Control System, Zhejiang University, Hangzhou 310000, China*
[2]*State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310000, China*
[3]*Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310000, China*

Correspondence should be addressed to Dongqin Feng; dqfeng@iipc.zju.edu.cn

Received 21 October 2015; Revised 27 January 2016; Accepted 15 February 2016

Academic Editor: Yongji Wang

With the development of industrial informatization, the industrial control network has gradually become much accessible for attackers. A series of vulnerabilities will therefore be exposed, especially the vulnerability of exclusive industrial communication protocols (ICPs), which has not yet been attached with enough emphasis. In this paper, stochastic game theory is applied on the vulnerability analysis of clock synchronization protocol (CSP), one of the pivotal ICPs. The stochastic game model is built strictly according to the protocol with both Man-in-the-Middle (MIM) attack and dependability failures being taken into account. The situation of multiple attack routes is considered for depicting the practical attack scenarios, and the introduction of time aspect characterizes the success probabilities of attackers actions. The vulnerability analysis is then realized through determining the optimal strategies of attacker under different states of system, respectively.

1. Introduction

The increasing interconnectivity of industrial control systems (ICS, as shown in Figure 1) exposes them to a wide range of vulnerabilities; the ICS now commonly accepts open standard protocols, bringing convenience to industrial automation. Yet these protocols also introduce vulnerabilities to an ICS.

These vulnerabilities are classified as two categories according to the position where they appear. Some mainly appear in the process control layer and information management layer (upper-middle layers in the industrial control network), such as the vulnerability of Internet interception and open OPC interface, which are mostly caused by introduction of traditional Internet technologies including TCP/IP technique and general operating system. However others mainly appear in field device layer (lower layer in the industrial control network) and refer to exclusive industrial communication protocols which are designed for ensuring real-time performance and stability rather than security.

As for the vulnerabilities in the upper layers, general network security technologies as firewall and access control are applied. There are several standards targeting both security assessment and security management. The ISO 15408 Common Criteria standard [1] specifies criteria for qualitative evaluation of the security level of a system, while ISO 13355 Guidelines for the management of IT security [2] provide guidelines on risk management of IT security. According to those standards, a high level description can be used to perform qualitative assessments of system properties, such as the security levels obtained by Common Criteria [3].

However, such methods focus upon evaluation of static behaviors of the system while ignoring the dependencies of events or time aspects of failures. Thus these methods cannot be used to predict in detail the behavior of ICS protocols and particularly for communications related to real-world intentional attack scenarios. Moreover, as mentioned in NIST-800 [4, 5], the vulnerability of exclusive industrial communication protocols is an open problem and mainly evolves the exclusive industrial communication protocols which are designed for ensuring real-time performance and stability rather than security. Thus, the research on the vulnerability analysis of clock synchronization protocol, one of the most important industrial communication protocols, is urgently needed.

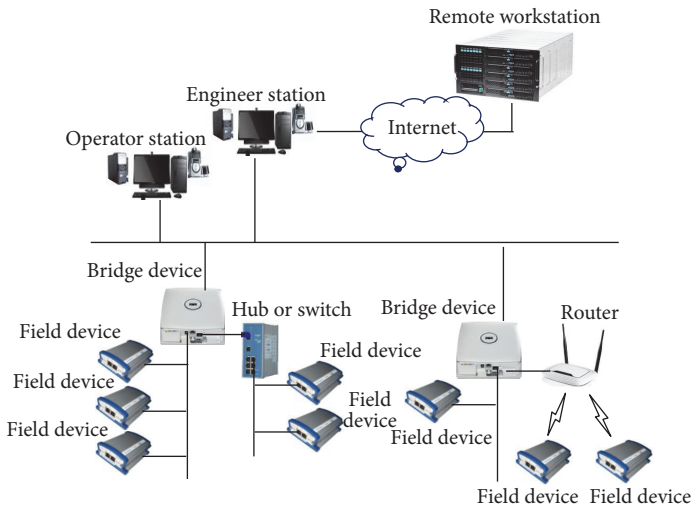

Figure 1: Typical framework of ICS (industrial control system).

Current research in the fields of vulnerability analysis of protocol involves the following analyses: from manual analysis to automatic analysis, from local analysis to integral analysis, and from rule-based analysis to model-based analysis. The methods used in the most popular articles include logic-based [6–11], theorem-based [12–17], and model detection-based [18–26] vulnerability analysis. Comparing with logic-based and theorem-based method, the model-based method obtains all possible actions and statuses of system through building a precise model in order to analyze vulnerabilities integrally. Unknown vulnerabilities can also be found out. Additionally building a model is relatively easier than extracting rules of the system. However, the model-based methods mainly involving stochastic model fail to depict the important character of vulnerability issues, the external malicious human factors (i.e., the decisions made by human) [27], which could be relatively readily described by game theory. Moreover, as pointed out in [28], vulnerability analysis must assume that an attackers' choice of action will depend on system state which may change over time. It indicates that attacker strategy depends on the CSP implementation. Hence, in this paper, we try to apply stochastic game theory in the vulnerability analysis of CSP. The exact model of CSP is built based on stochastic game theory with multiple attack routes and dependability issues included.

The paper is organized as follows. Section 2 introduces the related work in the vulnerability analysis of protocol and indicates and compares several methods of vulnerability analysis. CSP is introduced in Section 3; meanwhile the possible vulnerabilities that may be exploited by malicious behaviors are discussed. Then, the corresponding stochastic model of CSP including issues both of security and dependability is also built. In Section 4, basic concepts of stochastic game are introduced and then demonstrated by its application in CSP. Section 5 concludes the paper by summing up the main contribution of it and outlining the future work.

2. Related Works

Vulnerability analysis of protocol has been a research focus since the 1970s with a large scale of methods being proposed. According to the principle and development order, the methods can be divided into three kinds, namely, logic-based [6–11], theorem-based [12–17], and model detection-based [18–26] vulnerability analysis.

2.1. Logic-Based Method. The logic-based vulnerability analysis of protocol is the most intuitive, which has been shown to be effective and has discovered a number of protocol flaws. The logic-based vulnerability analysis of protocol involves the following steps:

(1) formalization of the protocol messages;

(2) specification of the initial assumptions;

(3) specification of the protocol goals;

(4) application of the logical postulates.

Formalization of the protocol messages involves specifying the protocol in the language of the logic by expressing each protocol message as a logical formula. The initial assumptions state the beliefs and possessions of protocol principals at the beginning of a protocol run and the protocol goals formalize the desired beliefs and possessions of principals after a successful protocol run. The objective of the logical analysis is to analyze whether the protocol goals can be derived from the initial assumptions and the formalized protocol by applying the logical postulates. If so, the protocol is robust; otherwise, the protocol is vulnerable.

Among the methods, the most representative one is Ban logic-based vulnerability analysis [6] proposed by Burrows et al. in 1989, which mainly focuses on the evolution of the belief in the implementation process of the protocol.

Using Ban logic-based method for vulnerability analysis will

(1) not make implicit assumptions;

(2) not take shortcuts;

(3) ensure thorough and unambiguous use of the postulates;

(4) not make implicit assumptions about failed goals;

(5) allow redundant assumptions to be identified easily.

The security flaws will then be identified rapidly and readily; nonetheless, the idealization and assumption involved in the process of Ban logic-based analysis will be prone to cause the threats such as leakage, modification, and forgery of the data.

As the extension, Ban-like logic including GNY logic [7, 8], AT logic [9], VO logic and SVO logic [10], and Kailar logic [11] can also be used to show how the beliefs of the trustworthy participants of the protocol evolve during the protocol runs and have better ability of describing the model. However, same as the Ban logic-based method, the Ban-like logics are incapable of proving the properties other than confidentiality, such as the correctness, the zero-knowledge, the real-time, and dependability.

2.2. Theorem-Based Method. Theorem-based method is used for proving the necessary security properties of corresponding protocol through theorem proof. Paulson [12–14] and Chadha et al. [15] proposed methods for proving security properties of protocols by induction, based on which Thayer et al. [16, 17] proposed the basic concepts of strand space. A strand is a sequence of events, which indicates either an execution by a legitimate party in a security protocol or else a sequence of actions by a penetrator, while a strand space is a collection of strands, which is equipped with a graph structure generated by causal interaction. Comparing with induction, the approach of strand space has the following advantages:

(1) Clear semantics to the assumption that certain data items such as nonces and session keys in security/authentication protocol are fresh and never arise in more than one protocol run.

(2) An explicit model of the possible behaviors of a system penetrator.

(3) Various notions of correctness.

(4) Proofs that are simple and informative.

However, the theorems and corresponding processes of proofing failed to be automatically described, which restricts the development of theorem-based method.

2.3. Model-Based Method. Model-based vulnerability analysis of protocol checks the security properties via state exploration. According to the difference of research path, it can be divided into forward research and backward research. In forward research, a state system is used for modeling the protocol with initial state being determined, and meanwhile a certain compromised state is set to be target state. The analysis begins from initial state, and the vulnerability of the protocol is determined by detecting the reachability of the target state. On the contrary, in backward research, the compromised state is regarded as the initial state, while the initial state is set to be the terminal state, the reachability of which determines the vulnerability of protocol as well.

Automated computational analysis tools are commonly used in model-based vulnerability analysis of protocol, while the protocol can be translated to the identifiable type through formal language. Famous computer scientist Hoare [18] designed the Communicating Sequential Process and corresponding model detection tool FDR (Failures Divergences Checker) for describing the information interaction in concurrent systems. Both CSP and FDR have been applied in analyzing NS protocol and other security protocols [19].

In addition, Dr. C. A. proposed Petri net in his Ph.D. thesis as a tool for modeling and analyzing concurrent system. The Petri net has following advantages:

(1) Strong describing ability, especially the Petri net with inhibitor arc which has the same describing ability with Turing machine.

(2) Graphical model, which is more intuitionistic to express the relationship of concurrence, sequence, conflict, synchronization, share, and so forth.

(3) Solid theoretical basis, many researchers have applied Petri net, CPN (Colored Petri net) in analyzing protocols since the 1990s [20]. Aura [21–23] adopted CPN for analyzing attackers behavior in several protocols, and then corresponding vulnerabilities of protocols were explored. Aura successfully analyzed the NS authentication protocol by using Predicate/Transition system. G.-S. Lee and J.-S. Lee [24] introduce time Petri net for analysis and assessment of cryptographic protocol. Reachability matrix of protocols states was built, and the reachability tree and attack sequences are then obtained. Crazzolara [25, 26] makes the vulnerability assessment of cryptographic protocol with the help of process net of Petri net, also known as process language.

2.4. Stochastic Game-Based Method. Among the three above-mentioned methods, comparing with logic-based and theorem-based method, model-based method is more suitable for accurately describing the states during the operation of protocol and quantitatively analyzing the vulnerability. Moreover, the introduction of graphic and automatic tools brings much convenience. Stochastic model is widely used in depicting the state transitions of protocol in the former studies which however ignore the description of the malicious behaviors implemented by the attacker. The state transitions of protocol under attack are unable to be reflected by only using stochastic model. Moreover, game theory is also a popular tool in the research field of vulnerability analysis for the reason that attacker and administrator can be viewed as players who are of contrary aims. The state transitions of protocol are therefore

the results of the interactions decided by the actions of both attacker and administrator. Nonetheless, the disadvantages in modeling ability, vivacity, and expansibility limit its application in the description and also the vulnerability analysis of the protocol.

As a combination, the stochastic game-based methods contain the advantages of both stochastic model and game theory. Based on the stochastic model, game theory can be introduced to correctly model intentional attacks upon a system and the attacker strategies are regarded as part of the set of transition probabilities between the states. There are increasing numbers of researches involving vulnerability analysis based on stochastic game. Syverson [29] analyzed the rational behaviors of both normal nodes and malicious nodes in the network based on stochastic game. Burke [30] built a model of attackers and defenders who are involved in an incomplete information repeated stochastic game. Lye and Wing [31] analyze the Nash equilibrium and optimal strategy of defender and attacker, respectively, based on stochastic model. In [32, 33], Wang et al. proposed a hierarchical stochastic game model which is applied to quantitatively analyze banking system and enterprise network. However the effect that variation of vulnerability index has on cost function is ignored. Most of related researches focus on the vulnerability analysis of traditional network system with the DoS and DDoS attack being considered. Nonetheless, given that the DoS and DDoS attack will be readily detected through the observation of anomalous load variation in the field bus. This kind of attack is rarely discussed in the context of malicious behavior aimed toward industrial communication protocol such as CSP. In addition, different from traditional network system, the transition of CSPs state follows the specified rules and also triggers conditions. In this paper, we apply the stochastic game on the vulnerability analysis of CSP; the model including states and transitions of which is specified strictly according to the protocol. Moreover, instead of DoS/DDoS attack which is commonly considered in most related research, MIM attack preferred by rational attackers is discussed with dependability failures such as hardware failure and software failure being taken into consideration as well. In addition, comparing with the model given in [34], we further consider the situation of multiple attack routes which is more appropriate for modeling the practical attack scenarios. The time aspect is also introduced in this paper for characterizing success probabilities of attackers actions, which is ignored in [31].

3. The Stochastic Model of CSP

Analogously to dependability analysis, we regard security breach states of CSP as failure states in the security community. In this paper, an attack toward CSP will therefore result from malicious behaviors which have been successful in exploiting existing vulnerabilities.

3.1. CSP and Malicious Behaviors. With the emergence of the IEEE 1588 PTP protocol, synchronous control of high precision becomes possible, which makes up for real-time

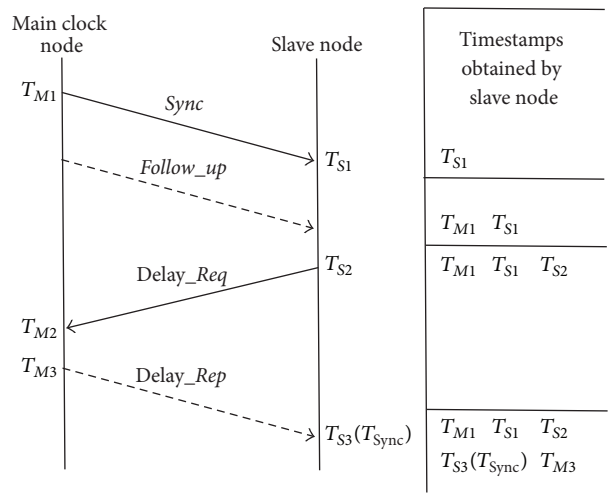

Figure 2: CSP without attack.

deficiencies in CSMA/CD mechanisms [35, 36], the main factor impeding the application of an Industrial Ethernet. State-of-the-art CSP can reach the level of microsecond and submicrosecond [37]. Processes of CSP without attacks are as shown in Figure 2. After obtaining specified timestamps, the slave node can calculate the value of $\text{Delay}_{\text{main\_slave}}$ and $\text{Offset}_{\text{main\_slave}}$ through formula (1) and formula (2), respectively. Then, the synchronized time of the slave node can be computed as T_{sync} from formula (3). Consider the following:

$$\text{Delay}_{\text{main\_slave}} = \frac{(T_{S1} - T_{M1}) + (T_{M2} - T_{S1})}{2}, \quad (1)$$

$$\text{Offset}_{\text{main\_slave}} = T_{S1} - T_{M1} - \text{Delay}, \quad (2)$$

$$T_{\text{sync}} = T_{M3} + \text{Offset} + \text{Delay}. \quad (3)$$

However, the corresponding vulnerabilities in such process could be easily acquired. Imagine that an attacker is able to intercept and even tamper with the Sync, Follow_up, Delay_Req and Delay_Rep clock synchronization command messages. Various kinds of attacks including Man-in-the-Middle (MIM), Denial of Service (DoS), and Freshness Attacks (FA), can be implemented due to the ignorance of confidentiality in CSP. Among these methods, MIM is preferred by rational attackers for the reason that all of $\text{Delay}_{\text{main\_attacker}}$, $\text{Offset}_{\text{main\_attacker}}$, $\text{Delay}_{\text{attacker\_slave}}$, and $\text{Offset}_{\text{attacker\_slave}}$ can be readily obtained. The main clock node will be completely spoofed while the slave node will be fully manipulated. A typical implementation of the MIM attack is as shown in Figure 3. The required timestamp information is collected during stage I. In stage II, attackers can fully master the real-time clock of slave clock node by sending bogus command messages while also preventing their detection by the main clock node.

3.2. The Stochastic Model. We model the expected time to exploit a specific vulnerability when using action a as negatively exponentially distributed in order to simplify

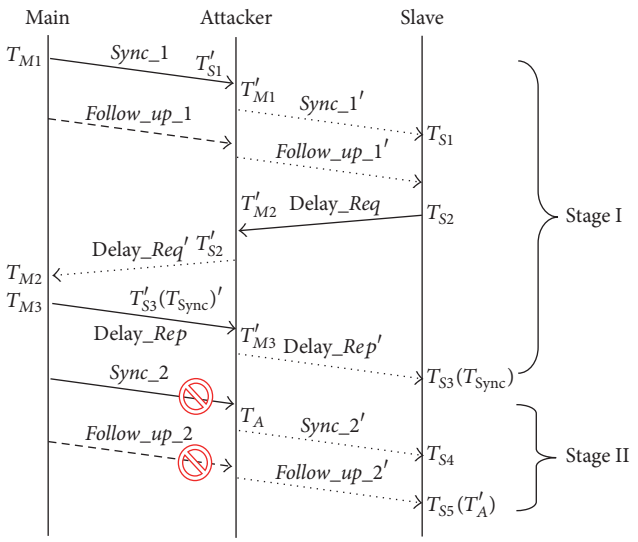

FIGURE 3: CSP under attack.

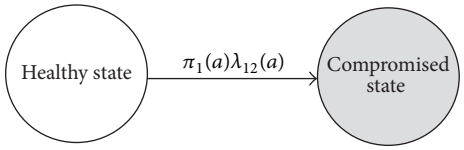

FIGURE 4: A two-state stochastic process with security failure rate assigned.

analytical assessment of the model. Rate $\lambda_{ij}(a)$, where i and j are two different states in the stochastic model, represents the expected time of transforming from state i to state j. In order to formalize the human-based decision factor, we define $\pi_i(a)$ as the probability that an attacker will choose action a when the system is in state i; this is almost the same as the method proposed in [38]. The vulnerability will be exploited when the system transforms from state i (health state of CSP) to state j (compromised state of CSP). Thus, the failure rate between states i and j may be computed as $q_{ij} = \pi_i(a) \cdot \lambda_{ij}(a)$ and as illustrated in Figure 4.

Remark 1. The states in stochastic model of CSP describe the specified situations of synchronization network, including the process of protocols implementation and the behaviors taken by the node devices (both normal ones and attackers) at that time. Consider the situation in which an attacker procures the main clock node and slave clock nodes configuration information (e.g., IP, MAC of both nodes); it can be viewed as a state.

Remark 2. The actions represent the attackers behaviors, which are identified according to the process described in Figure 3, and also the dependability failures including hardware failure and software failure. For example, when the attacker firstly receives the Sync_1 and Follow_up_1 from main clock node, he will choose to intercept, parse, and transmit them, which could be regarded as an action.

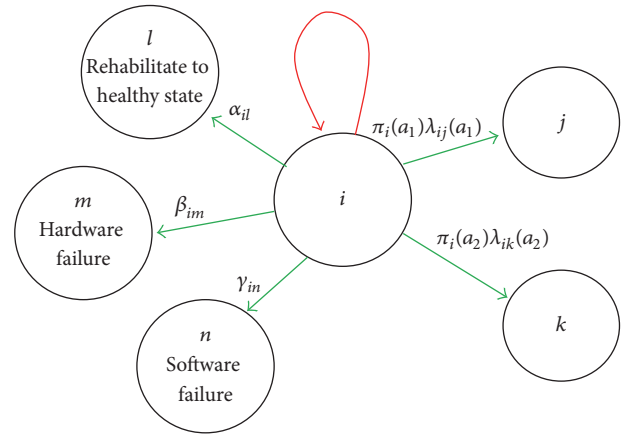

FIGURE 5: Stochastic process containing both security and dependability failure.

Meanwhile, the dependability failure could also be viewed as an action. More states and actions will be specified in the following chapters.

Remark 3. The rate here indicates the expected time of transforming from one state to another. Specifically, the security failure rate with respect to the attackers behaviors represents the expected time the attacker will spend on the transformation from healthy state to compromised state.

However, we consider not only security failures but also dependability failures such as hardware failure and software failure in that a security breach might also accidentally be caused by software bugs, hardware deterioration, administrative misconfiguration, and erroneous user input. By introducing both security failures and dependability failures, our model is made more realistic than the model given in [38]. The stochastic process, which incorporates security failures and dependability failures, is as shown in Figure 5. Note that, other than transiting to the several possible compromised states due to the security or dependability failures, the node still probably remains in the initially healthy state. Additionally, the attack toward CSP consists of many successive atomic attack actions and can therefore be modeled as a series of state changes, leading from an initially healthy state to one of several possible compromised states.

We then model the CSP under attack as a continuous-time Markov chain (CTMC) with a finite number of states $i = 1, \ldots, N$.

Let

$$X(t) = \{X_1(t), X_2(t), \ldots, X_N(t)\}, \tag{4}$$

where $X_i(t)$ denotes the probability that the system remains in state i at time t. The state equation describing all the intended and also unintended malicious behaviors toward CSP is then

$$\frac{d}{dt}X(t) = X(t)Q, \tag{5}$$

where Q is the $N \times N$ state transition rate matrix of the system. The element q_{ij} $(i \neq j)$ of Q is

$$q_{ij} = \lim_{dt \to 0} \left\{ \frac{\text{Prob}\left(\text{transition from } i \text{ to } j \text{ in } (t, t+dt)\right)}{dt} \right\}, \quad (6)$$

$$q_{ii} = -\sum_{j \neq i} q_{ij}.$$

Hence, in the example of Figure 4, the ith row in the transition rate matrix Q will be

$$Q_{i\cdot} \supseteq \left\{q_{ii}, q_{ij}, q_{ik}, q_{il}, q_{im}\right\} = \left\{-\left(\pi_i(a_1)\lambda_{ij} + \pi_i(a_2)\lambda_{ik} + \alpha_{il} + \beta_{im} + \gamma_{in}\right), \quad (7)$$
$$\pi_i(a_1)\lambda_{ij}, \pi_i(a_2)\lambda_{ik}, \alpha_{il}, \beta_{im}, \gamma_{in}\right\}.$$

Note that there will always be a possibility that an attacker does not choose any of the possible atomic attack actions a_1 and a_2, which means the attacker prefers to terminate the whole attack in order to obtain a greater reward; that is,

$$\pi_i(a_1) + \pi_i(a_2) + \pi_i(\emptyset) = 1. \quad (8)$$

Then, we regard A as a complete set of all possible atomic attack actions toward the system (including $\emptyset$). The strategy can be expressed as a sequence of actions that the attacker chooses. A complete attack strategy is denoted:

$$\prod = \{\pi_i,\ i = 1, 2, \ldots, N\}, \quad (9)$$

where N is the number of states the system might reach, and

$$\pi_i = \{\pi_i(a),\ a \in A\}. \quad (10)$$

π_i is the strategy vector for state i. Hence, $\pi_i(a)$ is the probability that the attacker will choose to perform action a in state i. We will also have

$$0 \leqslant \pi_i(a) \leqslant 1, \quad \forall i, a, \quad (11)$$

$$\sum_{\forall a \in A} \pi_i(a) = 1, \quad \forall i. \quad (12)$$

An attack action can be considered successful if the action causes an undesirable transformation of the current system state. The transition probabilities between states will therefore be an important aspect of the expected reward when an attacker decides upon an action. If the system is in state i, the next state of the system is determined by the embedded transition probabilities p_{ij}:

$$p_{ij} = \frac{q_{ij}}{\sum_{i \neq j} q_{ij}}, \quad j = 1, \ldots, N,\ i \neq j. \quad (13)$$

In states where there exist one or more actions available to the attacker, an alternative transition probability can be computed by conditioning on the chosen action. The conditioned transition probabilities, denoted by $p_{ij}(a)$, model the probability that an attacker succeeds with a particular attack action a, assuming that he does not perform two actions simultaneously.

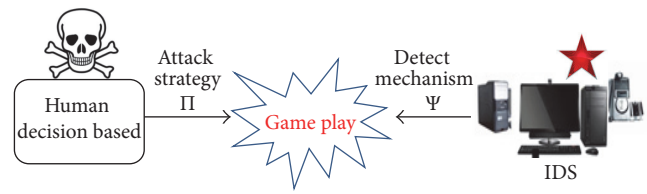

Figure 6: The game play between attacker and system.

For the example illustrated in Figure 5, we compute $p_{ij}(a_1)$ by inserting $\pi_i(a_1) = 1$ in the embedded transition probabilities in (14). Then

$$p_{ij}(a_1) = \frac{\lambda_{ij}}{\lambda_{ij} + \alpha_{il} + \beta_{im} + \gamma_{in}}. \quad (14)$$

$p_{ik}(a_2)$ could also be computed similarly. In this way, the dependability failure can be incorporated into the security failure.

4. The Stochastic Game Model

4.1. Basic Concepts of Stochastic Game. Based on the stochastic model we built before, we introduce game theory in order to create a generic and sound framework for computing the expected malicious behaviors of attackers. As a consequence, we decide to take advantage of the stochastic game theory mentioned in [39] as a mathematical tool. We regard each malicious action, which may cause a transition of the current system of CSP, as an action in a game where the attacker's choices of action are based on consideration of the possible consequences. The interactions between the attacker and the system itself can then be modeled as a game, as illustrated in Figure 6.

This stochastic game, in the context of security analysis, is usually regarded as a two-player, zero-sum, multistage game where, at each stage, the parameters of the game depend on the current state of the CTMC mentioned above.

The stochastic game can be defined as

$$\Gamma = \{\Gamma_i,\ i = 1, \ldots, m\}, \quad (15)$$

where Γ_i is the game element of state i. It is important to note that even though the state space of the CTMC may be very large, Γ will in general span only a subset of its states, those where an attacker is able to perform an atomic action.

Each game element Γ_i can be represented by an $n \times 2$ matrix:

$$\Gamma_i = \begin{pmatrix} \text{undetected} & \text{detected} \\ \vdots & \vdots \\ \mu_{i1}(a_m) & \mu_{i2}(a_m) \\ \vdots & \vdots \end{pmatrix}, \quad (16)$$

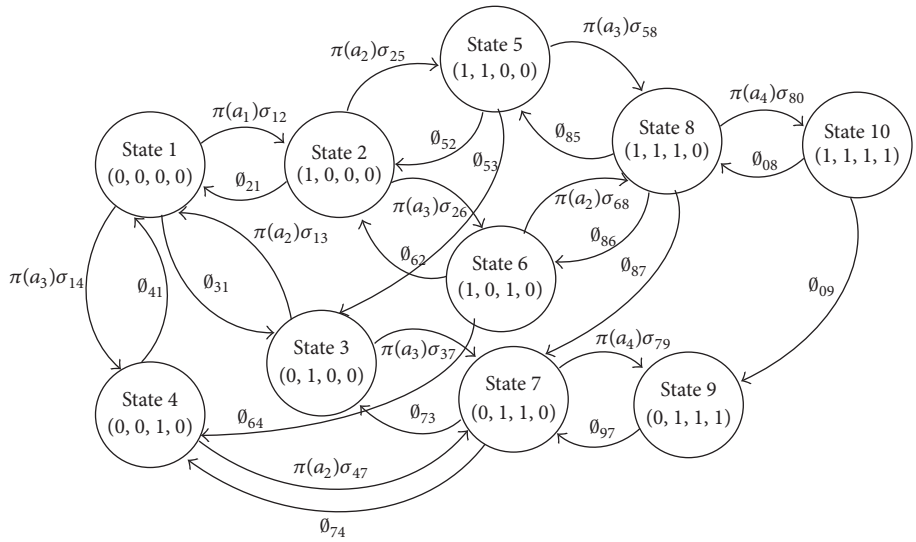

Figure 7: Security-related state transition diagram.

where n represents the number of possible atomic attack actions available to the attacker in state i. The elements in each row indicate the possible reward the attacker will receive by performing a specific attack action. The values of $\mu_{i1}(a_m)$ and $\mu_{i2}(a_m)$ can be computed as follows:

$$\mu_{i1}(a_m) = r_i(a_m \mid \text{undetected}) + \sum_{j=1,\ldots,z} p_{ij}(a_m)\,\Gamma_j,$$
$$\mu_{i2}(a_m) = r_i(a_m \mid \text{detected}). \tag{17}$$

The conditional transition probabilities $p_{ij}(a_m)$ can be obtained from formula (14). According to formula (17), if the attacker chooses the mth possible action in state i and the action remains undetected, the attacker will then receive the reward given by $r_i(a_m \mid \text{undetected})$. Moreover, the attacker will also receive an extra reward given by $\sum_{j=1,\ldots,z} p_{ij}(a_m)\Gamma_j$ for the reason that the attacker has to continue playing the jth game element. Meanwhile, when the attack action is detected, the attacker receives a nonpositive reward $r_i(a_m \mid \text{detected})$, and the game ends.

As shown in Figure 5, the detection mechanism performed by the system is denoted by Ψ, and let

$$\Psi = \{\theta_i,\ i = 1, \ldots, m\}, \tag{18}$$

where

$$\theta_i = \{\theta_i(a),\ a \in A\}. \tag{19}$$

θ_i is the probability that attack action a will be detected in state i. Through formulas (8), (15), and (17), we are able to compute the expected reward an attacker will receive for attack action a in state i:

$$E(\pi_i, \theta_i) = \sum_{\forall a \in A} \pi_i(a)\left(\left(1 - \theta_i(a)\right)\mu_{i1}(a) + \theta_i(a)\,\mu_{i2}(a)\right). \tag{20}$$

An attacker who tries to maximize the reward will choose the strategy $\pi_i^*(a)$ for each $a \in A$. Then the set of optimal strategies of the attacker $\prod^* = \{\pi_i^*,\ i = 1, \ldots, m\}$ can be obtained by iterative computation of Γ_j in formula (17). As a consequence, the maximal attack reward $\max E(\pi_i^*, \theta_i)$ $i = 1, \ldots, m$ can be readily computed.

4.2. Stochastic Game in CSP. In order to describe the available vulnerabilities of CSP, we define the set of steady state probabilities according to Figure 5 and formula (4):

$$X_i = \{i = \{\alpha, \beta, \lambda, \gamma\} \mid \alpha, \beta, \lambda, \gamma \in \{0, 1\}\}. \tag{21}$$

By formula (21), $i = (1, 0, 0, 0)$ represents that an attacker procures the source information (e.g., IP and MAC) of the main clock node and the slave clock node. State $i = (0, 1, 0, 0)$ denotes that the information of the main clock nodes real-time clock has been completely achieved by attacker, making it possible for the attacker to be synchronized with main clock node and yet not be detected. State $i = (0, 0, 1, 0)$ indicates that the attacker gains complete information of the slave clock node real-time clock and is capable of synchronizing the slave clock node. State $i = (0, 1, 1, 1)$ represents the situation where the attacker changes the real-time clock of slave clock node without being detected, after first obtaining information of both nodes and being synchronized with both nodes. We assume that attacker lacks the ability to change slave node real-time clock without being detected, in the case where the attacker does not have clock information of both nodes and is not synchronized with both nodes. It also means some states including $i = (0, 0, 0, 1)$, $i = (0, 0, 1, 1)$, $i = (0, 1, 0, 1)$, $i = (1, 0, 0, 1)$, $i = (1, 0, 1, 1)$, and $i = (1, 1, 0, 1)$ are unreachable in our model due to the fact that they are out of the step with the strict demands of time sequence in CSP, and if we build the model with every states reachable, the model will then become rather unrealistic and unconvincing.

The action set can be defined according to the process described in Figure 3:

$$A = \{a_1, a_2, a_3, a_4, \emptyset\}. \tag{22}$$

a_1 intercept, parse, and transmit Sync_1, Follow_up_1, in order to obtain T_{M1}, T'_{S1}, and T'_{M1}; a_2 intercept, parse Delay_req, Delay_rep, and transmit as Delay_req$'$ in order to obtain T'_{M2}, T'_{S2}, and T_{M2}; a_3 transmit Delay_rep$'$; a_4 intercept and block Sync_2, Follow_up_2, and then transmit Sync_2$'$, Follow_up_2$'$.

With timestamp information T_{M1}, T'_{S1}, T_{M2}, and T'_{S2}, we can compute $\text{Offset}_{\text{main\_attack}}$ and $\text{Delay}_{\text{main\_attacker}}$. The main clock node's real-time clock will be completely achieved by the attacker, making it possible for the attacker to be synchronized with the main clock node and not be detected. Meanwhile, with timestamp information T'_{M1}, T_{S1}, T'_{M2}, and T_{S2}, we can compute $\text{Offset}_{\text{attacker\_slave}}$ and $\text{Delay}_{\text{attacker\_slave}}$. Then the slave clock node will be synchronized with the attacker. By sending Sync_2$'$, Follow_up_2$'$, with attacker's own timestamp included, the attacker is capable of controlling the real-time clock of the slave clock node.

The attacker's priorities, rewards, and costs of actions are as shown in Table 1.

Following the analysis we made before, we obtain the security-related state transition diagram as illustrated in Figure 7, in which more than one attack route can be readily made use of by the attacker for arriving at final security breach states (state 10 and state 9), from initial secure state (state 1).

And then we define the attack and detection rate as shown in Table 2 according to a practical configuration in order to make the model more realistic and the results more convincing.

According to formulas (11), (13), and (14), our game elements will then be

$$\begin{aligned}
\Gamma_{(0,0,0,0)} &= \begin{pmatrix} 10 + 1.0 \cdot \Gamma_{(1,0,0,0)} & -10 \\ 20 + 1.0 \cdot \Gamma_{(0,1,0,0)} & -20 \\ 20 + 1.0 \cdot \Gamma_{(0,0,1,0)} & -20 \\ -5 & 0 \end{pmatrix}, \\
\Gamma_{(1,0,0,0)} &= \begin{pmatrix} 20 + 0.83 \cdot \Gamma_{(1,1,0,0)} & -20 \\ 20 + 0.83 \cdot \Gamma_{(1,0,1,0)} & -20 \\ -10 & 0 \end{pmatrix}, \\
\Gamma_{(0,1,0,0)} &= \begin{pmatrix} 20 + 0.9 \cdot \Gamma_{(0,1,1,0)} & -20 \\ -15 & 0 \end{pmatrix}, \\
\Gamma_{(0,0,1,0)} &= \begin{pmatrix} 20 + 0.9 \cdot \Gamma_{(0,1,1,0)} & -20 \\ -15 & 0 \end{pmatrix}, \\
\Gamma_{(1,1,0,0)} &= \begin{pmatrix} 20 + 0.77 \cdot \Gamma_{(1,1,1,0)} & -20 \\ -20 & 0 \end{pmatrix}, \\
\Gamma_{(1,0,1,0)} &= \begin{pmatrix} 20 + 0.77 \cdot \Gamma_{(1,1,1,0)} & -20 \\ -20 & 0 \end{pmatrix}, \\
\Gamma_{(0,1,1,0)} &= \begin{pmatrix} 30 + 0.9 \cdot \Gamma_{(0,1,1,1)} & -30 \\ -25 & 0 \end{pmatrix}, \\
\Gamma_{(1,1,1,0)} &= \begin{pmatrix} 30 + 0.83 \cdot \Gamma_{(0,1,1,1)} & -30 \\ -30 & 0 \end{pmatrix}, \\
\Gamma_{(0,1,1,1)} &= \Gamma_{(0,1,1,1)} = 30.
\end{aligned} \tag{23}$$

Solve the stochastic game according to formula (17) and compute $\max E(\pi_i^*, \theta_i)$ $i = 1, \ldots, m$:

$$\begin{aligned}
\Pi^* = \{&\pi^*_{(0,0,0,0)}, \pi^*_{(1,0,0,0)}, \pi^*_{(0,1,0,0)}, \pi^*_{(0,0,1,0)}, \pi^*_{(1,1,0,0)}, \\
&\pi^*_{(1,0,1,0)}, \pi^*_{(0,1,1,0)}, \pi^*_{(1,1,1,0)}\},
\end{aligned} \tag{24}$$

where

$$\begin{aligned}
\pi^*_{(0,0,0,0)} &= \{\pi^*_{(0,0,0,0)}(a_1), \pi^*_{(0,0,0,0)}(a_2), \pi^*_{(0,0,0,0)}(a_3), \\
&\quad \pi^*_{(0,0,0,0)}(\emptyset)\}, \\
\pi^*_{(1,0,0,0)} &= \{\pi^*_{(1,0,0,0)}(a_2), \pi^*_{(1,0,0,0)}(a_3), \pi^*_{(1,0,0,0)}(\emptyset)\}, \\
\pi^*_{(0,1,0,0)} &= \{\pi^*_{(0,1,0,0)}(a_3), \pi^*_{(0,1,0,0)}(\emptyset)\}, \\
\pi^*_{(0,0,1,0)} &= \{\pi^*_{(0,0,1,0)}(a_2), \pi^*_{(0,0,1,0)}(\emptyset)\}, \\
\pi^*_{(1,1,0,0)} &= \{\pi^*_{(1,1,0,0)}(a_3), \pi^*_{(1,1,0,0)}(\emptyset)\}, \\
\pi^*_{(1,0,1,0)} &= \{\pi^*_{(1,0,1,0)}(a_2), \pi^*_{(1,0,1,0)}(\emptyset)\}, \\
\pi^*_{(0,1,1,0)} &= \{\pi^*_{(0,1,1,0)}(a_4), \pi^*_{(0,1,1,0)}(\emptyset)\}, \\
\pi^*_{(1,1,1,0)} &= \{\pi^*_{(1,1,1,0)}(a_4), \pi^*_{(1,1,1,0)}(\emptyset)\}.
\end{aligned} \tag{25}$$

Take solving $\pi^*_{(1,1,1,0)}$ as an example,

$$\Gamma_{(1,1,1,0)} = \begin{pmatrix} 30 + 0.83 \cdot \Gamma_{(0,1,1,1)} & -30 \\ -30 & 0 \end{pmatrix}. \tag{26}$$

Assume that $\pi^*_{(1,1,1,0)}(a_4)$ is ω and $\pi^*_{(1,1,1,0)}(\emptyset)$ is therefore $1 - \omega$, then formula (17) would be

$$\begin{aligned}
E = \omega\left((1 - 0.3)(30 + 0.83 \cdot 30) + 0.3 \cdot (-30)\right), \\
\theta \in [0, 1].
\end{aligned} \tag{27}$$

It seems that $\omega = 1$ (namely, $\pi^*_{(1,1,1,0)} = \{1, 0\}$) is the solution of $\pi^*_{(1,1,1,0)}$; however, according to the basic concept of Nash equilibrium, each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only their own strategy. If each player has chosen a strategy and no player can benefit by changing strategies while the other players keep their unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash equilibrium.

In our paper, attacker and defender are viewed as two players. Thus, the Nash equilibrium equation is obtained

Table 1: Priorities, rewards, costs, and detection probabilities of attack actions.

Priority	Action	$R_i(a \mid \text{undetected})$	$R_i(a \mid \text{detected})$	$\theta_i(a)$
1	a_4	+30	−30	0.3
2	a_2, a_3	+20	−20	0.8
3	a_1	+10	−10	0.6
4	$\emptyset$	$r_{(0,0,0,0)}(\emptyset) = -5$ $r_{(1,0,0,0)}(\emptyset) = -10$ $r_{(0,1,0,0)}(\emptyset) = -15$ $r_{(0,0,1,0)}(\emptyset) = -15$ $r_{(1,1,0,0)}(\emptyset) = -20$ $r_{(1,0,1,0)}(\emptyset) = -20$ $r_{(0,1,1,0)}(\emptyset) = -25$ $r_{(1,1,1,0)}(\emptyset) = -30$	0	0

Table 2: The description of states.

State	Description (the information that the attacker obtains)
State 1	(1) None (initial healthy state)
State 2	(1) Configuration of both main clock node and slave node
State 3	(1) Real-time clock information of main clock node
State 4	(1) Real-time clock information of slave clock node
State 5	(1) Configuration of both main clock node and slave node (2) Real-time clock information of main clock node
State 6	(1) Configuration of both main clock node and slave node (2) Real-time clock information of slave clock node
State 7	(1) Real-time clock information of main clock node (2) Real-time clock information of slave clock node
State 8	(1) Configuration of both main clock node and slave node (2) Real-time clock information of main clock node (3) Real-time clock information of slave clock node
State 9	(1) Real-time clock information of main clock node (2) Real-time clock information of slave clock node (3) Attacker could change time without being detected
State 10	(1) Configuration of both main clock node and slave node (2) Real-time clock information of main clock node (3) Real-time clock information of slave clock node (4) Attack could change time without being detected

based on the condition that no matter being detected or not, the payoff of attacker would be the same. Namely, when attacker settles his strategy down, the defender cannot benefit by changing his own strategy. Noted that the benefit of defender is to decrease the payoff of attacker. The Nash equilibrium equation is as follows:

$$\omega \cdot (30 + 0.83 \cdot 30) + (1 - \omega)(-30) = (-30) \cdot \omega. \quad (28)$$

Then, $\omega = 0.26$ ($\pi^*_{(1,1,1,0)} = \{0.26, 0.74\}$) would be the solution of $\pi^*_{(1,1,1,0)}$, namely, the Nash equilibria of one element of the game, $\pi^*_{(1,1,1,0)} = \{0.26, 0.74\}$ means that, in state $i = (1, 1, 1, 0)$, the probability attacker will choose to perform action a_4 is 0.26, while that of not choosing to perform any action is 0.74. By following the strategy obtained from Nash equilibria of the game, the attackers are able to mitigate risk of being detected and maximize the payoff as possible. And we can then compute Π^* for the whole game through iteration of Γ.

Table 3: Optimal strategies for rational attacker.

State 1	$\pi^*_{(0,0,0,0)}$	(0, 0.064, 0.064, 0.872)
State 2	$\pi^*_{(1,0,0,0)}$	(0.114, 0.114, 0.772)
State 3	$\pi^*_{(0,1,0,0)}$	(0.296, 0.704)
State 4	$\pi^*_{(0,0,1,0)}$	(0.296, 0.704)
State 5	$\pi^*_{(1,1,0,0)}$	(0.371, 0.629)
State 6	$\pi^*_{(1,0,1,0)}$	(0.371, 0.629)
State 7	$\pi^*_{(0,1,1,0)}$	(0.245, 0.755)
State 8	$\pi^*_{(1,1,1,0)}$	(0.26, 0.74)

However, we need to note that $\pi^*_{(1,1,1,0)}$ is the best strategy for a rational attacker, and actually some risk ignorant attackers (also known as irrational attackers) will probably choose a totally different strategy. Under this circumstance, this is equivalent to setting $\theta_i(a) = 0$. The best strategy for this attacker would therefore be $\pi^*_{(1,1,1,0)} = \{1, 0\}$, and the method we propose still can be applied and the best strategies for irrational attacker can be obtained even much easier. We have not taken this situation into consideration because of its ease of being detected and thus less loss will then be caused comparing with the loss caused by rational attacker that we analyze in this paper.

Through the iteration of each Γ, the best strategy of a rational attacker in each state of the system can be computed as shown in Table 3. What is more, a bar graph which corresponds to Table 3 is also obtained, as shown in Figure 8.

$\pi^*_{(0,0,0,0)}$ means, in state 1, the attacker has four choices of action, namely, a_1, a_2, a_3, and $\emptyset$, which has been thoroughly explained in formula (22) and Table 1. The values (0, 0.064, 0.064, 0.872) mentioned in Table 3 are the probability of choosing each action. More specifically, in state 1, the probability of choosing a_1 is 0, that of choosing a_2 is 0.064, that of choosing a_3 is also 0.064, and that of choosing $\emptyset$ is 0.872. Hence, in state 1, the attacker tends to take no actions. $\pi^*_{(1,0,0,0)}$-$\pi^*_{(1,1,1,0)}$, the best strategy of attack in the states other than state 1, can also be explained in the same way.

Figure 8 is for vividly reflecting the results shown in Table 3. In Figure 8, the yellow bar represents the probability of attackers choosing taking no actions, while dark green and light green indicate the probability of attackers choosing action a_2 and a_3, respectively. The x-axis is divided by different states, while y-axis shows the values of probability.

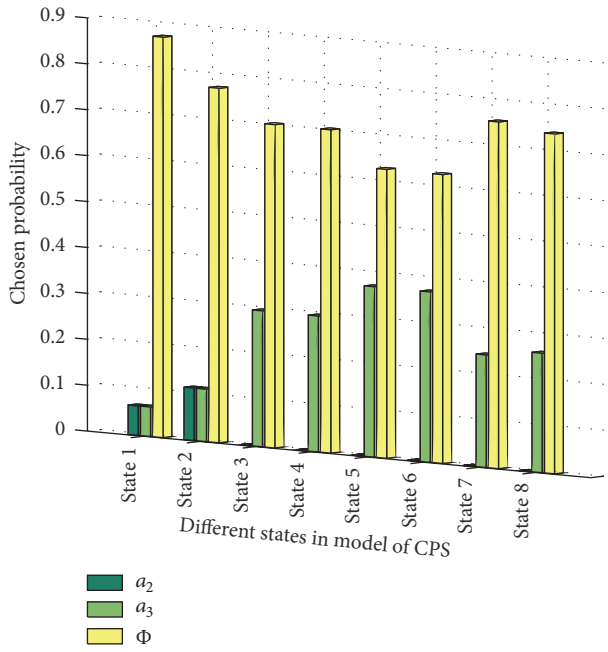

Figure 8: Optimal strategies for rational attacker.

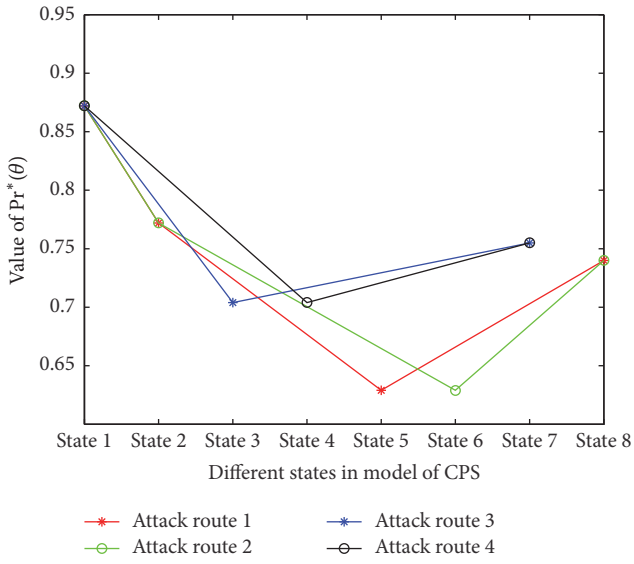

Figure 9: The value of $\Pr^*(\theta)$ in each step of all attacking routes.

In order to further analyze the attackers optimal strategy, the variation of $\Pr^*(\theta)$, namely, the probability of attackers taking no action (can also be viewed as attackers giving up the attack), is as shown in Figure 9. Additionally, based on the analysis of security-related state transition as shown in Figure 7, we then are able to obtain four attack routes (from initial healthy state, state 1, to final compromised state, state 7 or state 8) which are distinguished by different colors.

Table 4 is used for quantitatively explaining the variation in Figure 9, which would also more directly reflect the variation of attackers willing in different stages.

Table 4: Variation of $\Pr^*(\theta)$ in each step of all attack routes.

Attack routes	Each step	Variation of $\Pr^*(\theta)$
Attack route 1/2	State 1 to state 2	−11.47%
	State 2 to state 5/6	−18.5%
	State 5/6 to state 8	+17.6%
Attack route 3/4	State 1 to state 3/4	−19.26%
	State 3/4 to state 7	+7.24%

As shown in Figure 7, the red line indicates attack route 1, from state 1 to state 5 and then from state 5 to state 8. From state 1 to state 2, the probability decreases from 0.872 to 0.772, which reduces 11.47%. All of the parameters in the third column of Table 4 can be explained in the same way. Note that attack routes 1 and 2 and attack routes 3 and 4 are categorized into the same row, respectively, due to the same value of variation in each stage. We then are able to analyze the willingness of attacker in the different stages during the process of CSP and adopt appropriate countermeasures. For example, through comparing attack routes 1/2 with attack routes 3/4, the attacker would be more interested in implementing action a_2 or a_3 rather than a_1. As a consequence, the limited resource should be used on the protection of real-time clock information of main clock node or slave clock node. Specifically, we should take priority to encrypt the real-time clock field in the clock synchronization messages.

5. Concluding Remarks

In this paper, we demonstrate how to analyze malicious attacks upon a CSP using stochastic game theory. We modify the methods proposed in [17, 18, 20, 21] in order to make our model more accurate, realistic, and versatile. We not only introduce different attack routes and dependability failures, but also take into consideration the time aspect of attacks. CSP and malicious behaviors toward it are introduced. We then build the corresponding stochastic game model with several attack routes and dependability failures included. Finally, we obtain the optimal strategies of an attacker for the different states of the system.

In the future, we are interested to apply the method we propose to different kinds of industrial communication and several modifications may also be made. Moreover, the approach is based on the underlying assumption that the attackers have a complete overview of the system including states, transition rates, and detection rates, and the game is actually a zero-sum stochastic game; these might not always be valid assumptions. Thus, games of incomplete information and non-zero-sum games will therefore be another focus of our research. Additionally, according to the strategies of attacker, the administrator (defender) will no doubt make some pertinent changes in defense mechanism (e.g., different policies used in IDS), and the parameters in the game (e.g., detection rate) will also change. In consequence, the optimal strategy will be different. A rational attacker will not always

implement the same strategy. Under this circumstance, dynamic game will be much more suitable.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by National High Technology Research and Development Program (863 Project) of China under Grant 2012AA041102, in part by National Natural Science Foundation of China under Grant 61223004, and in part by National Natural Science Foundation of China under Grant 61433006.

References

[1] ISO, "Common criteria for information technology security evaluation," ISO 15408, 1999, http://www.commoncriteria.org/.

[2] ISO/IEC, "Information technology—guidelines for the management of IT security," ISO/IEC 13335, 1993, http://www.iso.ch.

[3] ISO 15408, Common Criteria for Information Technology Security Evaluation, 1999, http://www.commoncriteria.org.

[4] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," NIST Special Publication 800-82, National Institute of Standards and Technology, Gaithersburg, Md, USA, 2007.

[5] J. P. Wack, M. C. Tracy, and M. P. Souppaya, "Guideline on network security testing," *NIST Special Publication*, vol. 800, no. 42, pp. 13–14, 2003.

[6] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London A. Mathematical and Physical Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[7] L. Gong, R. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols," in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 234–248, IEEE, Oakland, Calif, USA, May 1990.

[8] S. H. Brackin, "A HOL extension of GNY for automatically analyzing cryptographic protocol," in *Proceedings of the 9th IEEE Workshop on Computer Security Foundations (CSFW '96)*, pp. 62–76, June 1996.

[9] M. Abadi and M. R. Tuttle, "A semantics for a logic of authentication," in *Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing*, pp. 201–216, ACM, Quebec, Canada, August 1991.

[10] P. F. Syverson and P. C. Van Oorschot, "On unifying some cryptographic protocol logics," in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy (SP '94)*, pp. 14–28, IEEE, 1994.

[11] R. Kailar, "Reasoning about accountability in protocols for electronic commerce," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 236–250, IEEE, Oakland, Calif, USA, May 1995.

[12] L. C. Paulson, "Proving properties of security protocols by induction," in *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, pp. 70–83, Rockport, Mass, USA, June 1997.

[13] L. C. Paulson, "Mechanized proofs for a recursive authentication protocol," in *Proceedings of the 10th IEEE Workshop on Computer Security Foundations (CSFW '97)*, pp. 84–94, Rockport, Mass, USA, June 1997.

[14] L. C. Paulson, "Inductive approach to verifying cryptographic protocols," *Journal of Computer Security*, vol. 6, no. 1, pp. 85–128, 1998.

[15] R. Chadha, M. Kanovich, and A. Scedrov, "Inductive methods and contract-signing protocols," in *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01)*, pp. 176–185, ACM, November 2001.

[16] F. J. T. Fabrega, J. C. Herzog, and J. D. Guttman, "Strand spaces: why is a security protocol correct?" in *Proceedings of the IEEE Symposium on Security and Privacy (SP '98)*, pp. 160–171, IEEE, Oakland, Calif, USA, May 1998.

[17] F. J. T. Fabrega, J. C. Herzog, and J. D. Guttman, "Strand spaces: proving security protocols correct," *Journal of Computer Security*, vol. 7, no. 2, pp. 191–230, 1999.

[18] C. A. R. Hoare, "Communicating sequential processes," *Communications of the ACM*, vol. 21, no. 8, pp. 666–677, 1978.

[19] G. Lowe, "Breaking and fixing the Needham-Schroeder Public-Key Protocol using FDR," in *Tools and Algorithms for the Construction and Analysis of Systems*, T. Margaria and B. Steffen, Eds., vol. 1055 of *Lecture Notes in Computer Science*, pp. 147–166, Springer, Berlin, Germany, 1996.

[20] B. B. Nieh and S. E. Tavares, "Modelling and analyzing cryptographic protocols using Petri nets," in *Advances in Cryptology—AUSCRYPT '92*, vol. 718 of *Lecture Notes in Computer Science*, pp. 275–295, Springer, Berlin, Germany, 1993.

[21] I. Al-Azzoni, D. G. Down, and R. Khedri, "Modeling and verification of cryptographic protocols using coloured Petri nets and Design/CPN," *Nordic Journal of Computing*, vol. 12, no. 3, pp. 201–228, 2005.

[22] A. Basyouni, *Analysis of Wireless Cryptographic Protocols*, Queen's University at Kingston, 1998.

[23] T. Aura, "Modelling the needham-schroeder authentication protocol with high level petri nets," *Helsinki University of Technology Digital Systems Laboratory Series B: Technical Reports*, vol. 14, no. 1, pp. 12–14, 1995.

[24] G.-S. Lee and J.-S. Lee, "Petri net based models for specification and analysis of cryptographic protocols," *Journal of Systems and Software*, vol. 37, no. 2, pp. 141–159, 1997.

[25] F. Crazzolara and G. Winskel, *Language, Semantics, and Methods for Cryptographic Protocols*, BRICS, Computer Science Department, University of Aarhus, Aarhus, Denmark, 2000.

[26] F. Crazzolara and G. Winskel, "Petri nets in cryptographic protocols," in *Proceedings of the IEEE 15th International Parallel & Distributed Processing Symposium (IPDPS '01)*, vol. 1, p. 149, 2001.

[27] K. B. B. Madan, K. Goševa-Popstojanova, K. Vaidyanathan, and K. S. Trivedi, "A method for modeling and quantifying the security attributes of intrusion tolerant systems," *Performance Evaluation*, vol. 56, no. 1–4, pp. 167–186, 2004.

[28] D. M. Nicol, W. H. Sanders, and K. S. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 48–64, 2004.

[29] P. F. Syverson, "Different look at secure distributed computation," in *Proceedings of the 10th IEEE Computer Security Foundations Workshop (CSFW '97)*, pp. 109–115, Rockport, Mass, USA, June 1997.

[30] D. A. Burke, *Towards a Game Theory Model of Information Warfare*, Air Force Institute of Technology, Dayton, Ohio, USA, 1999.

[31] K.-W. Lye and J. M. Wing, "Game strategies in network security," *International Journal of Information Security*, vol. 4, no. 1-2, pp. 71–86, 2005.

[32] Y. Wang, C. Lin, K. Meng, H. Yang, and J. Lv, "Security analysis for online banking system using hierarchical stochastic game nets model," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–6, IEEE, Honolulu, Hawaii, USA, December 2009.

[33] Y. Wang, C. Lin, Y. Wang, and K. Meng, "Security analysis of enterprise network based on stochastic game nets model," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, IEEE, Dresden, Germany, June 2009.

[34] B. K. Sallhammar and S. J. Knapskog, "Using game theory in stochastic models for quantifying security," in *Proceedings of the 9th Nordic Workshop on Secure IT-Systems (NordSec '04)*, Espoo, Finland, November 2004.

[35] X. Q. Miao, "Exposition of six type of communication protocols of real-time ethernet," *Process Automation Instrumentation*, vol. 26, no. 4, pp. 1–6, 2005.

[36] W. Yanshan, L. Yunhua, and L. Enpeng, "Research and application of Ethernet time synchronization," *Measurement and Control Technology*, vol. 26, no. 4, pp. 4–6, 2007.

[37] H. Weibel, "IEEE 1588 tutorial," in *Conference on IEEE*, vol. 1588, pp. 1–56, 2006.

[38] K. Sallhammar, S. J. Knapskog, and B. E. Helvik, "Using stochastic game theory to compute the expected behavior of attackers," in *Proceedings of the Symposium on Applications and the Internet Workshops (SAINT '05)*, pp. 102–105, February 2005.

[39] G. Owen, *Game Theory*, Academic Press, New York, NY, USA, 2nd edition, 1982.