

RESEARCH ARTICLE

Verification of Archive System Opacity With Bounded Labeled Petri Nets

ZHENZHONG LIU School of Economics and Finance, Xi'an Jiaotong University, Xi'an 710061, China
Shaanxi Xiao Baodang Mining Company Ltd., Yulin 719302, China

e-mail: liuzhenzhong1357@163.com

ABSTRACT Opacity is an essential security indicator in archive systems. There exists a set of secret states and an external intruder who can observe the behavior of the system in the archive system. The intruder can steal private information by observing the behavior of the system. The system is said to be K -step opaque when the intruder cannot confirm whether the system has been in a secret state at any time, within the observation of K events. In the case where an intruder can never be sure whether the system has ever been in a secret state, the system is referred to be infinite-step opaque. To be realistic, we consider an archive system modeled as a bounded labeled Petri net, and propose an algorithm for constructing a modified state estimator to increase the security of the archive system. Our aim is to verify the two types of opacity of the system by the observer and the modified state estimator. Our new algorithm improves the security of the system so that an intruder cannot easily know whether the system is in a secret state or not, which also improves the previously-known results.


INDEX TERMS Petri nets, discrete event systems, opacity, archive systems.

I. INTRODUCTION

Motivated by the concern about security and privacy in cyber-physical systems, opacity has been extensively investigated in the past years [1], [2], [3], [4], [5], [6], [7]. With the rapid development of information technology, electronic archival systems play a vital role in modern society. These systems are widely used to protect, manage, and provide access to large amounts of sensitive information. The archive systems mainly classify and preserve secret documents, where different documents have different levels of security, with a higher level representing a more critical document. Higher level secret documents in the archive system should be more strictly protected, increasing the system's robustness against attacks. Therefore, in order to make the research more in line with the real situation, this paper uses an archive system simulated by a labeled Petri net (LPN) as the research object to analyze the opacity problem of secret states with different safety levels in the system. However, with advances in information technology, archives systems are facing new security challenges, especially in protecting

secret information. Protecting secret information in archives systems is a critical task, and effective security measures must be put in place to prevent unauthorized access, tampering, or disclosure. Over the past decades, Petri nets have been widely used to model and analyze a variety of systems, including discrete-event systems. Petri nets are a graphical tool that can effectively describe the behavior and state changes of a system. However, the traditional Petri net model has some limitations when dealing with the secret information. Since the information flow in the traditional Petri net model is transparent, i.e., the information transfer path can be observed directly in the model. In some cases, this transparency may lead to the risk of leakage of secret information [8], [9], [10], [11], [12].

To address the challenge of protecting secret information within an archival system and to ensure that the system does not disclose sensitive data when processing the information, researchers have begun to use labeled Petri nets (LPNs) to verify opacity of the system. LPNs are an extension of traditional Petri nets that introduce the concept of labels that can be used to label and track the information in a system. By using LPNs, the works in [13] and [14] build a model in which the flow of information is opaque,

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamad Afendee Mohamed .

thus preventing direct observation of secret information. This approach improves the overall security and confidentiality of the archive system. The works in [15] and [16] analyze diagnosability and prognosticability of labeled Petri nets using the basis reachability graph.

The goal of opacity verification in an archive system is to ensure that the operation of the system is secure and correct while protecting the secret information. The main issue of opacity verification is to determine whether there is a sequence of actions in the system that could lead to the disclosure of the secret information. By performing opacity verification, the work in [17] identifies potential security vulnerabilities and information leakage paths, and proposes appropriate security measures to prevent these problems. In this way, the archive system can protect secret information more reliably and ensure that it can always meet security requirements during the operation.

As information interactions in networked systems continue to increase, to ensure information flow properties, opacity has become the latest area of concern for discrete event systems. Opacity reflects the ability of a system to hide a given secret from an intruder during the evolution of the system. The meaning of opacity is that it can formalize most of the properties of information flow. Thus, it cannot only formalizes security and privacy, but can also analyzes them together with three important properties of discrete event systems (DESS), i.e., detectability [18], diagnosability [19], and observability [20], [21]. Therefore, the purpose of this paper is to investigate how to verify opacity of archives systems by using LPNs. We focus on the protection of secret information and explore how to build a model to describe the behavior of the system. Through this research, we hope to provide an effective method for assessing the security level of archives systems and the level of protection of secret information. The ultimate goal is to provide stronger security for the design and implementation of archives systems to address the increasingly complex and diverse security threats.

The concept of K -step opacity was initially introduced by [22], where it was motivated by the analysis of cryptographic protocols. The notion of infinite-step opacity, an extension of K -step opacity, was later characterized by [23]. In the subsequent work [24], Saboori and Hadjicostis demonstrate that the verification of infinite-step opacity for a non-deterministic finite automaton can be achieved by constructing an observer and a bank of estimators, where each estimator represents a pair of potential initial state and current state. It is proven that verifying infinite-step opacity is PSPACE-hard. The work in [25] proposes an approach for checking both K -step opacity and infinite-step opacity, introducing a new structure called a two-way observer (TW-observer). The TW-observer is constructed by synchronizing the observer of a given automaton with the observer of its reversed automaton, known as the initial-state estimator. It is shown that the complexity of verifying both K -step opacity and infinite-step opacity is exponential in the number

of states of the system. Recently, Tong [26] et al. establish that these two opacity properties can be more efficiently verified by analyzing the states of the observer and the initial-state estimator, without constructing the TW-observer.

In this paper, we focus on verifying opacity in the archive system. Since there are various types of secret documents stored in the archive system and all of them have different levels of secrecy, we define multiple levels of secret states in the archive system and solve for multiple levels of opacity. We abstract the archive system as a bounded LPN system. Secrets are defined as a subset of reachable markings in the reachability graph, where secrets have several levels that represent different secret events in an archival system. The proposed method of opacity verification is based on the concept of basis reachability graph (BRG), which hides the unobservable events, reduces a large number of states, and generalizes the information in the system with a concise form. The size of the BRG is less than or equal to the reachability graph, based on the number of observable transitions. Therefore, the BRG is effectively used to verify certain opacity properties [27], [28].

The contributions of this paper can be summarized as follows.

(1) An enhanced opacity verification approach is provided in this paper. The opacity levels of different secret states are different, and we verify the opacity of each secret state respectively.

(2) A state estimator construction scheme for different levels of secret states is proposed, which establishes a stricter protection strategy for high-level secret states. We construct a modified state estimator model to increase the opacity of the secret states in order to prevent the leakage of high-level secret state information.

(3) The first enhanced opacity verification in a real-world scenario is presented, by modeling it through an actual archive system.

II. PRELIMINARIES

A. BASICS OF PETRI NETS

A Petri net is denoted as $N = (P, T, Pre, Post)$, where P is a set of places representing states or conditions in the system, T is a set of transitions representing events or actions that can occur in the system, $Pre: P \times T \rightarrow \mathbb{N}$ and $Post: P \times T \rightarrow \mathbb{N}$ denote the *pre*- and *post*-incidence functions, respectively, where \mathbb{N} is a set of non-negative integers. The incidence matrix is represented as $C = Post - Pre$.

A marking refers to a vector $M: P \rightarrow \mathbb{N}$, where each place is assigned a non-negative integer number representing the number of tokens. The marking of a place p is indicated by $M(p)$. A Petri net system $\langle N, M_0 \rangle$ consists of a Petri net N with its initial marking M_0 .

A transition t is enabled at marking M if $M \geq Pre(\cdot, t)$, allowing it to fire and result in a new marking $M' = M + C(\cdot, t)$. We use $M[\sigma]$ to indicate that the sequence of

transitions $\sigma = t_1 \cdots t_k$ is enabled at the marking M , and $M[\sigma]M'$ to denote that firing σ leads to M' . The set of all enabled transition sequences in a Petri net N from the marking M_0 is denoted as $L(N, M_0) = \{\sigma \in T^* \mid M_0[\sigma]\}$, where T^* is the Kleene-closure of T [29]. For a given transition sequence $\sigma \in T^*$, the function $\pi: T^* \rightarrow \mathbb{N}^n$ associates the firing vector $y = \pi(\sigma) \in \mathbb{N}^n$, where $y(t) = k$ represents that the occurrence of transition t is k times in σ . The length of σ is denoted as $|\sigma|$.

For simplicity, $\langle N, M_0 \rangle$ can be represented as a Petri net system. A marking M is reachable if there exists a transition sequence σ with $M_0[\sigma]M$. The reachability set $R(N, M_0)$ encompasses all markings that can be reached from M_0 . A Petri net system is regarded as bounded if there exists a non-negative integer $k \in \mathbb{N}$, satisfying the condition $M(p) \leq k$ for any place $p \in P$ and any reachable marking $M \in R(N, M_0)$. The sets of input transitions and that of output transitions of a marking M are denoted by $T(\bullet M) = \{t \in T \mid M'[t]M, M' \in R(N, M_0)\}$ and $T(M\bullet) = \{t \in T \mid M[t]M', M' \in R(N, M_0)\}$. The set of input transitions for a set of markings \mathcal{M} is denoted by $T(\bullet \mathcal{M}) = \{t \in T \mid M'[t]M, M' \in \mathcal{M}\}$, the output transitions set of a set of markings is the same as the input set.

An LPN system can be represented as a four-tuple $G = (N, M_0, E, \ell)$, where N denotes a Petri net, M_0 denotes the initial marking, E denotes the alphabet (a set of labels), and $\ell: T \rightarrow E \cup \{\varepsilon\}$ is the labeling function that assigns symbols from E or the empty word ε to each transition $t \in T$. Consequently, the set of transitions can be divided into two distinct sets: $T = T_o \cup T_u$, where $T_o = \{t \in T \mid \ell(t) = E\}$ denotes a set of observable transitions, and $T_u = T \setminus T_o$ is denoted as a set of unobservable transitions. We denote the cardinality of observable transitions as $n_o = |T_o|$ and the cardinality of unobservable transitions as $n_u = |T_u|$. For a given marking $M \in R(N, M_0)$, $U(M)$ is defined as the unobservable reach, which refers to the set of markings reachable from M through unobservable transitions, which is denoted as $U(M) = \{M' \in \mathbb{N}^m \mid M[\sigma_u]M', \sigma_u \in T_u^*\}$. Additionally, if Y is a subset of markings in $R(N, M_0)$, $U(Y) = \bigcup_{M \in Y} U(M)$.

The labeling function can be extended to transition sequences, denoted as $\ell: T^* \rightarrow E^*$. This extension is defined as $\ell(\sigma t) = \ell(\sigma)\ell(t)$, where $\sigma \in T^*$ and $t \in T$. The language generated by G , represented as $\mathcal{L}(G)$, which is denoted as $\mathcal{L}(G) = \{\omega \in E^* \mid \exists \sigma \in L(N, M_0) : \omega \in \ell(\sigma)\}$. The words represented by $\mathcal{L}(G)$ can be observed by the intruder. An observed word $\omega \in \mathcal{L}(G)$ is referred to as an observation. We denote the set of markings consistent with the observation ω as $\mathcal{C}(\omega) = \{M \in \mathbb{N}^m \mid \exists \sigma \in L(N, M_0) : M_0[\sigma]M, \ell(\sigma) = \omega\}$.

Let $G = (N, M_0, E, \ell)$ be an LPN system. We define the T_u -induced subnet $N_u = (P, T_u, Pre_u, Post_u)$ as the net obtained by removing transitions in $T \setminus T_u$ from the Petri net N , where Pre_u and $Post_u$ represent the restrictions of Pre and $Post$ to T_u , respectively. The incidence matrix of the T_u -induced subnet is denoted as $C_u = Post_u - Pre_u$.

B. BASIS MARKINGS

In this subsection, we provide a brief overview of basis markings and basis reachability graph. For a comprehensive understanding of these concepts, we recommend referring to [30] for further details.

Definition 1: Given a marking M and an observable transition $t \in T_o$, we define

$$\sum(M, t) = \{\sigma \in T_u^* \mid M[\sigma]M', M' \geq Pre(\cdot, t)\}$$

as the set of explanations of t at M . In other words, it represents the collection of unobservable transition sequences that, when fired at M , enable the transition t . We define

$$Y(M, t) = \{y_u \in \mathbb{N}^{n_u} \mid \exists \sigma \in \sum(M, t) : y_u = \pi(\sigma)\}$$

as the corresponding set of e -vectors or firing vectors, associated with unobservable transition sequences in $\sum(M, t)$. Upon firing an unobservable transition sequence in $\sum(M, t)$ at M , transition t becomes enabled. In order to obtain a concise representation of the reachability set, our focus lies in identifying explanations with minimal firing vectors.

Definition 2: Given a marking M and an observable transition $t \in T_o$, we define

$$\begin{aligned} \sum_{\min}(M, t) \\ = \left\{ \sigma \in \sum(M, t) \mid \nexists \sigma' \in \sum(M, t) : \pi(\sigma') \preceq \pi(\sigma) \right\} \end{aligned}$$

as the set of minimal explanations of t at M , and we define

$$Y_{\min}(M, t) = \left\{ y_u \in \mathbb{N}^{n_u} \mid \sigma \in \sum_{\min}(M, t) : y_u = \pi(\sigma) \right\}$$

as the corresponding set of minimal e -vectors. Let $G = (N, M_0, E, \ell)$ be an LPN system. The BRG generated by a Petri net is a four-tuple $\mathcal{B} = (\mathcal{M}_b, \mathcal{T}_b, \delta_b, M_0)$, representing a non-deterministic finite state automaton comprised of all basis markings, where (1) the set \mathcal{M}_b represents all basis markings; (2) the set \mathcal{T}_b represents the set of transitions $t \in (T_o \cup T(\bullet \mathcal{M}_b))$, where \mathcal{M}_b is the set of secret states; (3) the transition function is denoted as $\delta_b: \mathcal{M}_b \times \mathcal{T}_b \rightarrow 2^{\mathcal{M}_b}$, i.e., $\delta_b(M_1, t) = M_2, M_2 = M_1 + C_u \cdot y_u + C(\cdot, t), y_u \in Y_{\min}(M_1, t)$. The function δ_b can be extended to $\mathcal{M}_b \times \mathcal{T}_b^* \rightarrow 2^{\mathcal{M}_b}$; (4) the state M_0 is the initial marking.

III. K-STEP OPACITY AND INFINITE-STEP OPACITY

The concepts of K -step opacity and infinite-step opacity have been defined in the framework of LPNs. In this paper, in order to apply to the LPN with multiple different levels of secret state, we improve the traditional state estimator. Within the realm of LPNs, a subset of reachable markings known as \mathcal{M}_S , where $\mathcal{M}_S \subseteq R(N, M_0)$, encapsulates the set of secrets. Specifically, any marking $M \in \mathcal{M}_S$ can be identified as a secret marking.

The main goal of this paper is to construct a modified state estimator to verify opacity in archive systems, where several secret states exist. First, we construct the initial system using

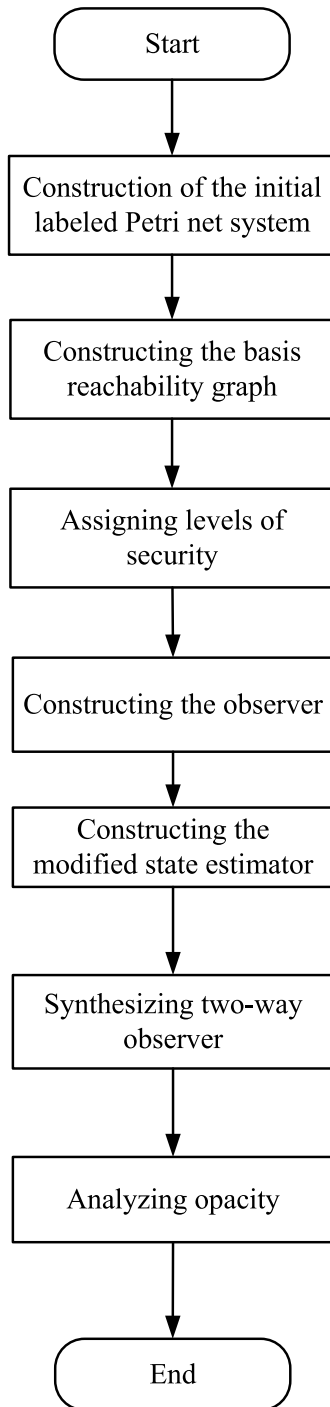


FIGURE 1. The flowchat of the scheme.

an LPN. Then, we simplify the scale of the system using the BRG, leaving only observable transitions. Next, we assign security levels in the system. Finally, we construct an observer and a modified state estimator to re-synthesize the two-way observer to verify the opacity of the secret state. The flowchart of the scheme is shown in Fig. 1. Given an LPN system $G = (N, M_0, E, \ell)$, the reachability graph of the LPN system is denoted by $G_R = (\mathcal{M}_R, M_0, E, f, \ell)$, \mathcal{M}_R is a

finite set of states, $f : \mathcal{M}_R \times E \rightarrow 2^{\mathcal{M}_R}$ is the transition function. We denoted $G_I = (\mathcal{M}_R, M_0, E, f_I, \ell)$ as the reversed reachability graph of G , where $\mathcal{M}_R = R(N, M_0)$. Particularly, the transition function $f_I : \mathcal{M}_R \times E \rightarrow 2^{\mathcal{M}_R}$ is defined by: for any two states M_1 and M_2 in \mathcal{M}_R and an event $e \in E$, we have $M_2 = f(M_1, e)$ if $M_1 \in f_I(M_2, e)$. Note that G_I is nondeterministic, and it is possible to reach several different states from one event. Additionally, the initial state of G_I is defined as the entire state space \mathcal{M}_R , given that a string in G_R may terminate at any state $M \in \mathcal{M}_R$. Then, for any string $s = \sigma_1\sigma_2 \dots \sigma_{|s|} \in E^*$, we denoted by s_I the reversed string of s , i.e., $s_I = \sigma_{|s|}\sigma_{|s|-1} \dots \sigma_1$.

We define a variant of opacity with reference to [21], which is formally defined in Definition 3 and Definition 4. We assume that the intruder, represented as an observer, possesses complete knowledge of the system's structure. However, it can only partially observe the system's behavior.

Definition 3: (*K*-step opacity). Given an LPN system $G = (N, M_0, E, \ell)$, the reachability graph $G_R = (\mathcal{M}_R, M_0, E, f, \ell)$, a set of secret states \mathcal{M}_S , and a non-negative integer $K \in \mathbb{N}$, the natural projection $\mathcal{P} : T^* \rightarrow T_o^*$, system G is said to be *K*-step opaque if

$$\begin{aligned}
 &(\forall st \in \mathcal{L}(G_R, M_0) : f(M_0, s) \in \mathcal{M}_S \wedge |\mathcal{P}(t)| \leq K) \\
 &(\exists s't' \in \mathcal{L}(G_R, M_0)) \\
 &[f(M_0, s') \notin \mathcal{M}_S \wedge \mathcal{P}(s') = \mathcal{P}(s) \wedge \mathcal{P}(t') = \mathcal{P}(t)].
 \end{aligned}$$

When $K = 0$, *K*-step opacity reduces to current-state opacity. When $K \rightarrow \infty$, *K*-step becomes infinite-step opacity. We recall the formal definition from [26].

Definition 4: (*Infinite-step opacity*). Given an LPN system $G = (N, M_0, E, \ell)$, a reachability graph $G_R = (\mathcal{M}_R, M_0, E, f, \ell)$, a set of secret states \mathcal{M}_S , and a non-negative integer $K \in \mathbb{N}$, system G is said to be infinite-step opaque if

$$\begin{aligned}
 &(\forall st \in \mathcal{L}(G_R, M_0) : f(M_0, s) \in \mathcal{M}_S) \\
 &(\exists s't' \in \mathcal{L}(G_R, M_0)) \\
 &[f(M_0, s') \notin \mathcal{M}_S \wedge \mathcal{P}(s') = \mathcal{P}(s) \wedge \mathcal{P}(t') = \mathcal{P}(t)].
 \end{aligned}$$

Definition 5: (*Multi-level opacity*). Given an LPN system $G = (N, M_0, E, \ell)$, a reachability graph $G_R = (\mathcal{M}_R, M_0, E, f, \ell)$, a secret requirement is a function $S : \mathcal{M}_R \rightarrow \mathbb{N}$ that assigns to each state $M \in \mathcal{M}_R$ a secret level $S(M)$.

Example 1: Consider an LPN system G shown in Fig. 2, representing an archive system for storing secret documents. In this LPN, the place p_2 represents the storage location where the secondary secret documents are stored. The remaining places are storage locations for documents classified at the first level of secret, with the second level of secret having a higher level of secret.

The set of observable transitions is $T_o = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9\}$ and the set of unobservable transitions is $T_u = \{t_{10}, t_{11}, t_{12}\}$. The set of observable transitions T_o represents operations of requesting secret documents such as borrowing, backing up, transferring, etc., which can be observed externally. On the contrary, while the set of unobservable transitions T_u belongs to internal operations

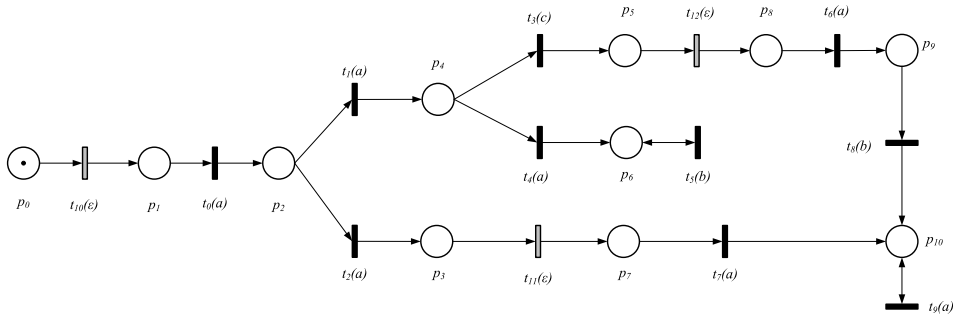


FIGURE 2. The LPN of the archive system.

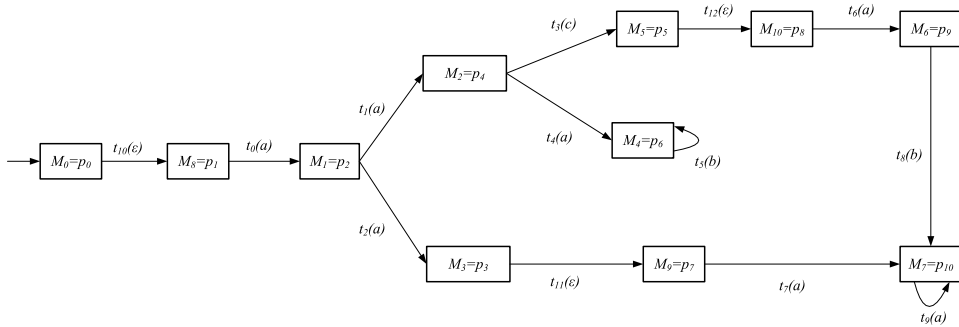


FIGURE 3. The reachability graph of the archive system.

on secret documents such as encrypting and decrypting documents, which cannot be observed externally. Transitions t_{10}, t_{11}, t_{12} are labeled as ϵ to indicate unobservable events, the rest of events are observable events.

As shown in Fig. 2, a secret document first enters into the first-level secret document storage location p_0 , arrives at the first-level secret storage location p_1 after the initialization operation t_{10} of the archive system, after a secret document transmission process t_0 reaching the first-level secret storage location p_2 , then the first-level secret storage location p_3 and the second-level secret storage location p_4 are reached after the document encryption operation t_1 and t_2 . The system arrives at the first-level secret storage locations p_5 and p_6 after the document request operations t_3 and t_4 , respectively. Next, after document backup operations t_{11} and t_{12} to reach the first level of secret storage locations p_7 and p_8 , then through document access operations to reach the first level of classified storage locations p_9 and p_{10} , and finally through document sealing processing t_9 to reach the first level of classified storage location p_{10} .

The reachability graph of the archive system is shown in Fig. 3. In the archive system, each state is supposed to be protected, with a secret level of at least 1, where M_2 is set to secret level 2. The BRG of the system is shown in Fig. 4. It is easy to verify that K -step opacity does not hold for $K = 2$. By taking $s = t_0$ and $t = t_4 t_1$, we know that the only string $s't' \in \mathcal{L}(G)$ is st itself. Note that the LPN system is not 2-step opaque.

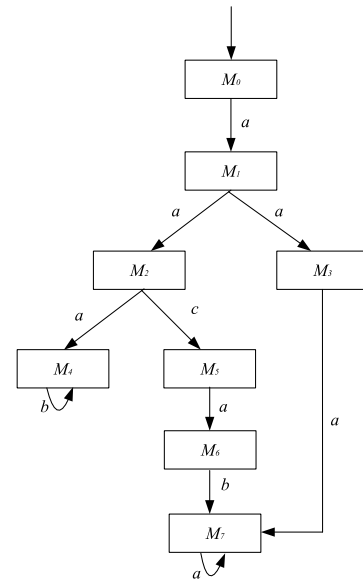


FIGURE 4. The basis reachability graph of the archive system.

IV. OPACITY VERIFICATION USING MODIFIED STATE ESTIMATOR

In this section, we present the results of verifying the two opacity properties using the BRG and the modified state estimator. The usage of BRG requires the satisfaction of the following two assumptions, which are common to most of the

literature in this area [30], [31], [32], [33]: (1) the LPN system is bounded; (2) its T_u -induced subnet is acyclic.

Given an LPN system $G = (N, M_0, E, \ell)$, the BRG is a nondeterministic finite automaton (NFA), where each state is a basis marking, the set of events is the alphabet of the LPN system, and there is no transition labeled with the empty word. We denote it as $G_B = (\mathcal{M}_B, E, f, M_0)$. It is proven that $\mathcal{L}(G_B) = \mathcal{L}(G)$. To avoid repeating material already presented in other works, we refer the readers to [32] for the algorithm to construct the BRG.

Given a BRG $G_B = (\mathcal{M}_B, E, f, M_0)$, we denote: $\mathcal{B}_{obs} = (\mathcal{M}_{obs}, E, f_{obs}, M_{obs,0})$ the observer of \mathcal{B} , and $\mathcal{B}_e = (\mathcal{X}_e, E, f_e, X_{e,0})$ the initial-state estimator of \mathcal{B} . The observer is constructed as shown in Algorithm 1.

Algorithm 1 Constructing an Observer for the Archive System

Input: An LPN $G = (N, M_0, E, \ell)$.
Output: A system observer $\mathcal{B}_{obs} = (\mathcal{M}_{obs}, E, f_{obs}, M_{obs,0})$.

- 1: Let $\mathcal{M} = \emptyset$, $\mathcal{M}_{new} = \{M_0\}$;
- 2: Let T_o be a set of observable events;
- 3: **while** $\mathcal{M}_{new} \neq \emptyset$ **do**
- 4: Select a state $M \in \mathcal{M}_{new}$;
- 5: **for all** $t \in T_o$ **do**
- 6: Computing $Y_{min}(M, t)$;
- 7: **for all** $y \in Y_{min}(M, t)$ **do**
- 8: Let $\hat{M} = M + C_{uc} \cdot y + C(\cdot, t)$;
- 9: **if** $\hat{M} \in \mathcal{M} \cup \mathcal{M}_{new}$ **then**
- 10: Let $\mathcal{M}_{new} = \mathcal{M}_{new} \cup \{\hat{M}\}$;
- 11: **end if**
- 12: Let $f(M, t) = \hat{M}$;
- 13: **end for**
- 14: **end for**
- 15: Let $\mathcal{M} = \mathcal{M} \cup \{M\}$;
- 16: Let $\mathcal{M}_{new} = \mathcal{M}_{new} \setminus \{M\}$;
- 17: **end while**
- 18: Let BRG $\mathcal{B} = (\mathcal{M}_B, E, f, M_0)$;
- 19: Generating an observer $\mathcal{B}_{obs} = (\mathcal{M}_{obs}, E, f_{obs}, M_{obs,0})$;
- 20: **Output** $\mathcal{B}_{obs} = (\mathcal{M}_{obs}, E, f_{obs}, M_{obs,0})$.

We denote by $\hat{X}(s, G_R)$ the current-state estimate associated with observed string $s \in \mathcal{P}(\mathcal{L}(G_R))$ w.r.t. G_R , i.e., $\hat{X}(s, G_R) = \{M \in \mathcal{M}_R : \exists t \in \mathcal{L}(G), f(M_0, t) = x \wedge \mathcal{P}(t) = s\}$.

In the archive system that is vulnerable to external intruders, we improve the state estimator of the system in order to resist the damage of the attack on the system. Each state in the archive system has a different level of confidentiality, and we should take stricter protection measures for high-level secret states. Therefore, we propose a scheme to improve the state estimator of the system.

In Algorithm 1, we combine the method of constructing observers with the basis reachability graph (BRG), omitting unobservable events and retaining observable events, which drastically reduces the number of states in the system. The

number of observer states grows exponentially with respect to the number of basis markings in the reachability graph.

The modified state estimator is constructed as shown in Algorithm 2. First, we construct a reversed automaton of the observer. Specifically, the transition function $f_R : \mathcal{M}_B \times E \rightarrow \mathcal{M}_B$ is defined by: for any two states M_1 and M_2 in \mathcal{M}_B and an event $e \in E$, we have $M_2 = f(M_1, e)$ if $M_1 \in f_R(M_2, e)$.

Algorithm 2 Constructing a Modified State Estimator of a System

Input: Archives system observer $\mathcal{B}_{obs} = (\mathcal{M}_{obs}, E, f_{obs}, M_{obs,0})$.
Output: State estimator for the archive system $\mathcal{B}_e = (\mathcal{M}_e, E, f_e, M_{e,0})$.

- 1: Let $G_a = (P, T_a, F_a, M_0, W_a)$;
- 2: Let $M_{e,0} = M_0$;
- 3: **for all** $M \in \mathcal{M}_o, t \in T_o$ **do**
- 4: **if** $f(M, t) = M'$ **then**
- 5: Let $f_e(M', t) = M$;
- 6: **end if**
- 7: **end for**
- 8: **for all** $M \in \mathcal{M}_e$ **do**
- 9: **if** $S(M) > 1$ **then**
- 10: Let $D = S(M)$
- 11: **for all** $1 < D$ **do**
- 12: **for all** $t' \in T(\bullet M)$ **do**
- 13: **if** $t \in D(M)$ **then**
- 14: Deleting the arc to which event t is connected;
- 15: **end if**
- 16: **end for**
- 17: $D = D - 1$;
- 18: **end for**
- 19: **end if**
- 20: **end for**
- 21: **Output** $\mathcal{B}_e = (\mathcal{M}_e, E, f_e, M_{e,0})$.

In the state estimator G_R , if a state M has a secret level $S(M) > 1$, we find the number of paths $D(M)$ from the initial set of states $\mathcal{M}_{R,0}$ to the state M . When $D \geq S(M)$, we delete $S(M) - 1$ events from the set of input events $T(\bullet M)$, and delete $D - 1$ events from the input events $T(\bullet M)$ when $1 < D < S(M)$. In both cases, if the secret state has more than one antecedent event in the state estimator, we choose the event with the shortest path length to delete.

Since in a real archive system, there are several secret documents of different levels in a real archive system, in order to make the proposed method more relevant to the real world, we propose a modified state estimator as shown in Algorithm 2. Then, there are multiple secret states in an archive system, the modified state estimator takes into account the policy that higher level secret states should be more strictly protected. Specifically, we remove some of the input events for states with the secret level greater than one.

Thus, it will be more difficult for an attacker to estimate the system behavior, which improves the security of the system.

For the complexity analysis of the algorithms, first, the complexity of constructing a BRG is exponential with respect to the size of the Petri network system (the number of places and the initial marking). Since both Algorithm 1 and Algorithm 2 are constructed under the BRG, in the worst case, constructing the BRG has the same complexity as constructing an RG, where the complexity is exponential related to the number of places and the initial marking. However, in practice, $|\mathcal{M}_B|$ is much smaller than $|R(N, M_0)|$. The complexity of the Algorithm 1 is of $O(|T_o| \times 2^{|\mathcal{M}_B|})$. Since Algorithm 2 is computed for secret states, it has the complexity of $O(|T_o| \times |\mathcal{M}_S| \times 2^{|\mathcal{M}_B|})$.

Definition 6: Given an LPN system $G = (N, M_0, E, \ell)$, the two-way observer of G is a deterministic finite-state automaton $Obs_{TW}(G) = (\mathcal{M}_{TW}, E_{TW}, f_{TW}, Q_{TW,0})$ where $\mathcal{M}_{TW} \subseteq \mathcal{M}_o \times \mathcal{M}_e$ is the set of states; $E_{TW} = (E \times \{\varepsilon\}) \cup (\{\varepsilon\} \times E)$ is the set of events; $Q_{TW,0} = (M_0, M_{e,0})$ is the initial state; $f_{TW}: \mathcal{M}_{TW} \times E_{TW} \rightarrow \mathcal{M}_{TW}$ is the transition function defined by: for any state $(M_1, M_2) \in \mathcal{M}_{TW}$ and event $e \in E$, the following transitions are defined whenever they are feasible

$$f_{TW}((M_1, M_2), (e, \varepsilon)) = (f(M_1, e), M_2)$$

$$f_{TW}((M_1, M_2), (\varepsilon, e)) = (M_1, f_e(M_2, e))$$

Intuitively, the TW-observer tracks a string s in $\mathcal{L}(G)$ from \mathcal{M}_B , and a reversed string t_R in $Rev(\mathcal{L}(G))$ from $M_{e,0}$, where $Rev(L) = \{s_R : s \in L\}$. Let (M_1, M_2) be a state reached in $Obs_{TW}(G)$. If $M_1 \cap M_2 \neq \emptyset$, then it represents that the two strings s and s_R are coincident in some states.

In a TW-observer, a reversed automaton infers the secret state of a system by monitoring and analyzing its inputs and outputs. When using a reversed automaton for opacity verification, we can modify the state transition diagram to protect the secret state in the system. Specifically, we can make modifications to the state transition diagram of the reversed automaton by adding or removing states and transitions, in order to conceal the actual states and transitions, thereby preventing attackers from inferring the secret state of the system based on the output of the reversed automaton. When designing the reversed automaton, we can also consider introducing secret states and events, and employ techniques such as encryption or obfuscation to safeguard the secret information in the system. In this way, even if attackers manage to obtain the output of the reversed automaton, they would be unable to deduce the secret information in the system. The main purpose of performing these modifications is to mask the true character of the system, thus preventing a potential attacker from inferring the secret state based on the output of the inverse automaton.

In the scheme of using BRG for opacity verification, we construct an initial archive system by using an LPN. By using a TW-observer, the reversed automaton can be flexibly modified, and the TW-observer can be reconstructed.

Our goal is to remove the input events of secret states in the BRG, which is equivalent to removing the output events of secret states in the reversed automaton. Ultimately, this achieves the satisfaction of 2-step opacity that was previously unsatisfied, or the satisfaction of 3-step opacity if it was previously satisfied with 2-step opacity. In the case of multiple paths, the shortest path from the initial state set of the reversed automaton is selected for deletion. In the archive system, to reflect the secrecy and security of the archive, we assume that the secrecy level of each state is at least 1. Based on protecting secret states, we construct the initial BRG system and ensure the security of secret states in the BRG. By using the BRG, we can obtain the observer \mathcal{B}_o and the reversed observer \mathcal{B}_e , and delete the input transitions of secret states in the reversed observer.

Example 2: In the archive system shown in Fig. 4, we classify the levels of secret states, establishing a distinction between the levels of secret. Specifically, we assign M_2 and M_3 are the level 2 secret states, which implies higher sensitivity and secrecy, while all other states are classified as level 1 secret states. The system observer \mathcal{B}_{obs} is shown in Fig. 5. The traditional state estimator is shown in Fig. 6. The modified state estimator generated after Algorithm 2 is shown in Fig. 7, where the pre-events are removed since M_2 and M_3 are secret states of level 2. The TW-observer synthesized by the observer and the state estimator is shown in Fig. 8, where only some of the necessary states are shown.

First, we consider the traditional state estimator with M_2 as the secret state, where $\mathcal{M}_S = \{M_2\}$. With the TW-observer, we find that state (C, J) is reached after string $\{(\varepsilon, a), (a, \varepsilon), (a, \varepsilon)\}$. Since the state can be represented

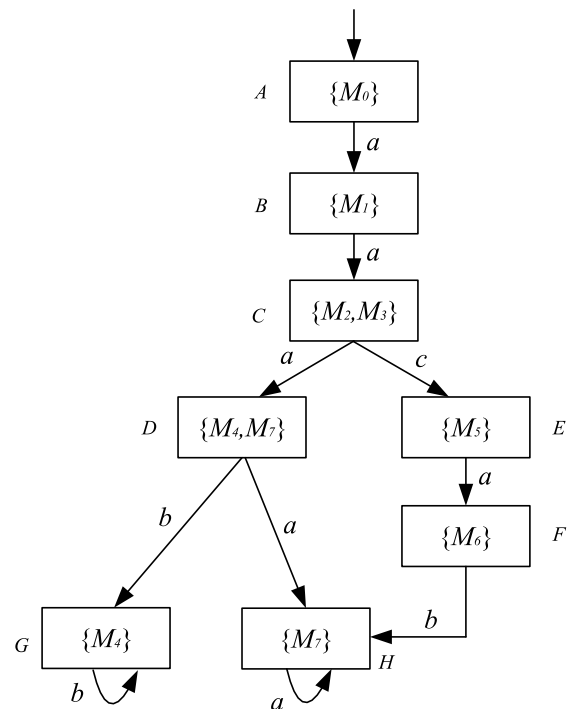


FIGURE 5. The observer \mathcal{B}_{obs} of the system.

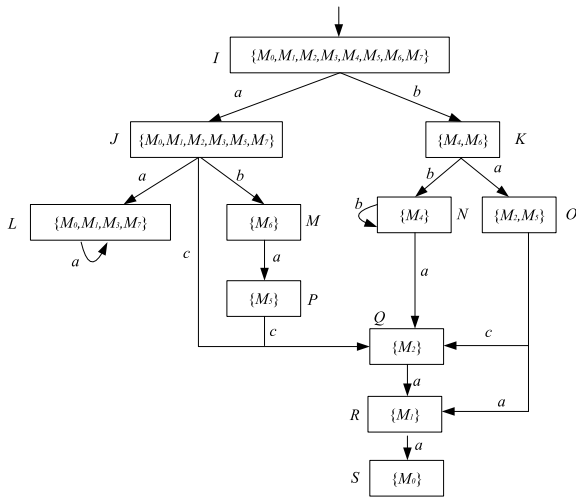


FIGURE 6. The traditional state estimator \mathcal{B}_e^t .

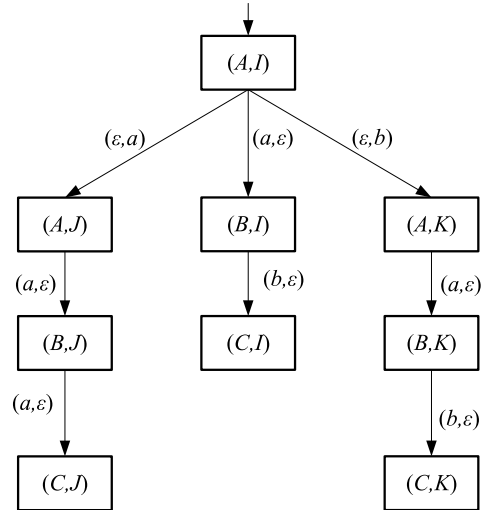


FIGURE 8. The reduced TW-observer of the system Obs_{TW} .

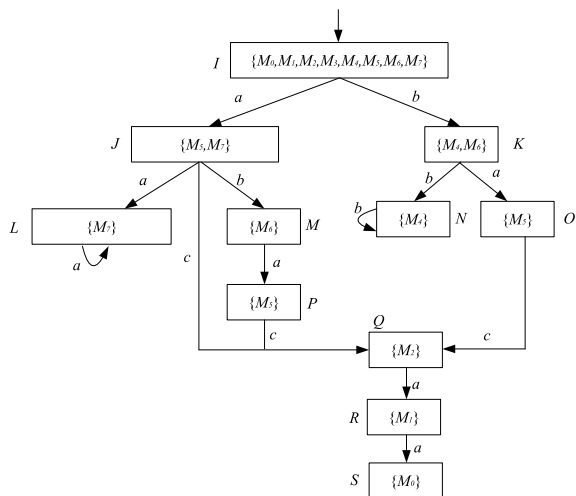


FIGURE 7. The modified state estimator \mathcal{B}_e^m .

$(\{M_2, M_3\}, \{M_0, M_1, M_2, M_3, M_5, M_7\})$, we have that $\{M_2, M_3\} \cap \{M_0, M_1, M_2, M_3, M_5, M_7\} = \{M_2, M_3\} \neq \mathcal{M}_S$. Therefore, the system satisfies 1-step opacity, and by the same logic it does not satisfy 2-step opacity.

Then, we consider the modified state estimator with M_2 as the secret state. Similarly, we find that state (C, J) is reached after string $\{(\epsilon, a), (a, \epsilon), (a, \epsilon)\}$. On the contrary, the state is $(\{M_2, M_3\}, \{M_5, M_7\})$, we have that $\{M_2, M_3\} \cap \{M_5, M_7\} = \emptyset \neq \mathcal{M}_S$. Therefore, the system satisfies 1-step opacity and 2-step opacity, by the same logic it does not satisfy 3-step opacity.

Finally, we consider the M_3 as the secret state. With the traditional state estimator, the system is not 2-step opaque, and with the modified state estimator, the system satisfies both 2-step opacity and infinite step opacity.

To strengthen our defenses against potential information leakage, we have carefully designed a modified state estimator that serves as a powerful barrier against unauthorized access to high-level secret information. Recognizing the

critical importance of protecting secret information, our approach focuses on introducing a higher level of sophistication for any potential intruder attempting to observe or decipher secret information. The modified state estimator significantly increases the difficulty for intruders to access secret information within the system. By hiding input events with high-level secret states, it is equivalent to adding an extra layer of defense, the possibility of leaking secret information is greatly minimized.

V. CONCLUSION

In this paper, the problem of opacity validation in archival systems is considered. To be realistic, we classify the secret states of the system into several levels, with higher levels supposed to be more strictly protected. The key to our approach is building an observer and a modified state estimator using BRG, which are designed to reduce the complexity associated with the verification process. The modified state estimator plays a key role in strengthening the system against potential intrusion risks. Specifically, the input events of states with secret level greater than 1 are hidden. This hiding strategy is crucial in reducing the vulnerability of the system, thus enhancing the overall security posture of the system. Then, we synthesize the TW-observer by using the observer and the modified state estimator, which is used to verify the system opacity, and the results show that the system can be changed from the original unsatisfied K -step opacity to satisfy the K -step opacity. With such a scheme, the system with several different levels of secret states can be verified in opacity, the higher level states being more strongly protected. In the future, we will consider the problem of opacity verification under different kinds of attacks [34] and consider how to reduce the computational complexity.

REFERENCES

- [1] J. Tan, F. Liu, and Z. Dziog, "Active opacity of discrete-event systems," *Int. J. Control*, vol. 96, no. 8, pp. 2090–2099, Aug. 2023.

- [2] X. Li, C. N. Hadjicostis, and Z. Li, "Extended insertion functions for opacity enforcement in discrete-event systems," *IEEE Trans. Autom. Control*, vol. 67, no. 10, pp. 5289–5303, Oct. 2022.
- [3] S. Habbachi, A. Zaghdoud, Z. Li, N. Wu, and M. Khalgui, "Secret inference and attackability analysis of discrete event systems," *Inf. Sci.*, vol. 609, pp. 1221–1238, Sep. 2022.
- [4] Z. Ma, J. Jiang, and K. Cai, "Secret protections with costs and disruptiveness in discrete-event systems using centralities," *IEEE Trans. Autom. Control*, Oct. 2024, doi: [10.1109/TAC.2023.3323531](https://doi.org/10.1109/TAC.2023.3323531).
- [5] M. Abdelmalak, V. Venkataramanan, and R. Macwan, "A survey of cyber-physical power system modeling methods for future energy systems," *IEEE Access*, vol. 10, pp. 99875–99896, 2022.
- [6] Z. Yu, X. Duan, X. Cong, X. Li, and L. Zheng, "Detection of actuator enablement attacks by Petri nets in supervisory control systems," *Mathematics*, vol. 11, no. 4, p. 943, Feb. 2023.
- [7] L. Huang, "Stability of cyber-physical systems of numerical methods for stochastic differential equations: Integrating the cyber and the physical of stochastic systems," *IEEE Access*, vol. 10, pp. 99479–99497, 2022.
- [8] J. Jang, K. Kim, S. Yoon, S. Lee, M. Ahn, and D. Shin, "Mission impact analysis by measuring the effect on physical combat operations associated with cyber asset damage," *IEEE Access*, vol. 11, pp. 45113–45128, 2023.
- [9] Y. Dong, Z. Li, and N. Wu, "Symbolic verification of current-state opacity of discrete event systems using Petri nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 12, pp. 7628–7641, Dec. 2022.
- [10] Z. Ma and K. Cai, "Optimal secret protections in discrete-event systems," *IEEE Trans. Autom. Control*, vol. 67, no. 6, pp. 2816–2828, Jun. 2022.
- [11] R. Jacob, J.-J. Lesage, and J.-M. Faure, "Overview of discrete event systems opacity: Models, validation, and quantification," *Annu. Rev. Control*, vol. 41, pp. 135–146, 2016.
- [12] M. Amin, F. F. M. El-Sousy, G. A. A. Aziz, K. Gaber, and O. A. Mohammed, "CPS attacks mitigation approaches on power electronic systems with security challenges for smart grid applications: A review," *IEEE Access*, vol. 9, pp. 38571–38601, 2021.
- [13] F. Basile, G. De Tommasi, and C. Motta, "Assessment of initial-state-opacity in live and bounded labeled Petri net systems via optimization techniques," *Automatica*, vol. 152, Jun. 2023, Art. no. 110911.
- [14] G. Zhu, Z. Li, and N. Wu, "Online verification of K -step opacity by Petri nets in centralized and decentralized structures," *Automatica*, vol. 145, Nov. 2022, Art. no. 110528.
- [15] N. Ran, T. Li, Z. He, and C. Seatzu, "Codiagnosability enforcement in labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 68, no. 4, pp. 2436–2443, Apr. 2023.
- [16] N. Ran, J. Hao, and C. Seatzu, "Prognosability analysis and enforcement of bounded labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 67, no. 10, pp. 5541–5547, Oct. 2022.
- [17] A. Labeled, I. Saadaoui, N. Wu, J. Yu, and Z. Li, "Current-state opacity verification in discrete event systems using an observer net," *Sci. Rep.*, vol. 12, no. 1, p. 21572, Dec. 2022.
- [18] K. Zhang, "Polynomial-time verification and enforcement of delayed strong detectability for discrete-event systems," *IEEE Trans. Autom. Control*, vol. 68, no. 1, pp. 510–515, Jan. 2023.
- [19] M. V. S. Alves, R. J. Barcelos, L. K. Carvalho, and J. C. Basilio, "Robust decentralized diagnosability of networked discrete event systems against DoS and deception attacks," *Nonlinear Anal. Hybrid Syst.*, vol. 44, May 2022, Art. no. 101162.
- [20] X. Cong, M. P. Fanti, A. M. Mangini, and Z. Li, "Critical observability of labeled time Petri net systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 3, pp. 2063–2074, Jul. 2023.
- [21] Y. Tong, H. Lan, and C. Seatzu, "Verification of K -step and infinite-step opacity of bounded labeled Petri nets," *Automatica*, vol. 140, Jun. 2022, Art. no. 110221.
- [22] A. Saboori and C. N. Hadjicostis, "Notions of security and opacity in discrete event systems," in *Proc. 46th IEEE Conf. Decis. Control*, Dec. 2007, pp. 5056–5061.
- [23] A. Saboori and C. N. Hadjicostis, "Verification of infinite-step opacity and analysis of its complexity," *IFAC Proc. Volumes*, vol. 42, no. 5, pp. 46–51, Jun. 2009.
- [24] A. Saboori and C. N. Hadjicostis, "Coverage analysis of mobile agent trajectory via state-based opacity formulations," *Control Eng. Pract.*, vol. 19, no. 9, pp. 967–977, Sep. 2011.
- [25] X. Yin and S. Lafortune, "A new approach for the verification of infinite-step and K -step opacity using two-way observers," *Automatica*, vol. 80, pp. 162–171, Jun. 2017.
- [26] A. Saboori and C. N. Hadjicostis, "Verification of infinite-step opacity and complexity considerations," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1265–1269, May 2012.
- [27] S. Habbachi, Z. Li, N. Wu, and M. Khalgui, "Language-based opacity verification and enforcement in the framework of labeled Petri nets," *Sci. Prog.*, vol. 105, no. 1, Feb. 2022, Art. no. 003685042210754.
- [28] F. Basile, G. De Tommasi, C. Motta, and C. Sterle, "Necessary and sufficient condition to assess initial-state-opacity in live bounded and reversible discrete event systems," *IEEE Control Syst. Lett.*, vol. 6, pp. 2683–2688, 2022.
- [29] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 3rd ed. Cham, Switzerland: Springer, 2021.
- [30] Z. Ma, Y. Tong, Z. Li, and A. Giua, "Basis marking representation of Petri net reachability spaces and its application to the reachability problem," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1078–1093, Mar. 2017.
- [31] M. P. Cabasino, A. Giua, M. Poggi, and C. Seatzu, "Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems," *Control Eng. Pract.*, vol. 19, no. 9, pp. 989–1001, Sep. 2011.
- [32] Y. Tong, Z. Li, C. Seatzu, and A. Giua, "Verification of state-based opacity using Petri nets," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2823–2837, Jun. 2017.
- [33] X. Cong, M. P. Fanti, A. M. Mangini, and Z. Li, "Critical observability verification and enforcement of labeled Petri nets by using basis markings," *IEEE Trans. Autom. Control*, vol. 68, no. 12, pp. 8158–8164, Dec. 2023.
- [34] Z. Yu, H. Gao, X. Cong, N. Wu, and H. B. Song, "A survey on cyber-physical systems security," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 21670–21686, Dec. 2023.



ZHENZHONG LIU received the M.S. degree in business administration from Xi'an University of Science and Technology, in 2015. He is currently pursuing the Ph.D. degree in applied economics with Xi'an Jiaotong University. His research interest includes efficiency and security evaluation of digital transformation in archives management.