*mathematics*

MDPI

*Article*

# Critical Observability Enforcement in Discrete Event Systems Using Differential Privacy

**Jie Zhang and Zhiwu Li ***

Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau SAR, China
* Correspondence: zwli@must.edu.mo

**Abstract:** In the context of discrete event systems (DESs), critical states usually refer to a system configuration of interest, describing certain important system properties, e.g., fault diagnosability, state/language opacity, and state/event concealment. Technically, a DES is critically observable if an intruder can always unambiguously infer, by observing the system output, whether the plant is currently in a predefined set of critical states or the current state set is disjointed with the critical states. In this paper, given a partially observable DES modeled with a finite-state automaton that is not critically observable, we focus on how to make it critically observable, which is achieved by proposing a novel enforcement mechanism based on differential privacy (DP). Specifically, we consider two observations where one observation cannot determine whether a system is currently in the predefined critical states (i.e., the observation violating the critical observability) while the other is randomly generated by the system. When these two observations are processed separately by the differential privacy mechanism (DPM), the system generates an output, exposed to the intruder, that is randomly modified such that its probability approximates the two observations. In other words, the intruder cannot determine the original input of a system by observing its output. In this way, even if the utilized DPM is published to the intruder, they are unable to identify whether critical observability is violated.

**Keywords:** discrete event system; critical observability; differential privacy

**MSC:** 93C65

check for updates

**Citation:** Zhang, J.; Li, Z. Critical Observability Enforcement in Discrete Event Systems Using Differential Privacy. *Mathematics* **2024**, *12*, 3842. https://doi.org/10.3390/math12233842

Academic Editor: António Lopes

Received: 10 November 2024
Revised: 26 November 2024
Accepted: 28 November 2024
Published: 5 December 2024

**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Over the past decade, state estimation in discrete event systems (DESs) (without a full sensor deployment) has received unprecedented interest thanks to the extensive deployment of cyber-physical systems that are mathematically characterized by discrete event systems. Within this problem, several core challenges, such as opacity analysis [1–3], diagnosability assessment [4,5], detectability verification [6,7], and controller design optimization [8], are deeply dependent on the ability to accurately estimate the system states by its partially observed nature. Particularly, critical observability, as an important concern in the direction of state estimation, is the focus of our exploration in this research. The concept of critical observability, formulated in [9], is widely applied to networked systems, medical and health systems, semiconductor manufacturing, etc. A system is considered to be critically observable if, through the observation of a system output, it is possible to infer without ambiguity whether or not the current system states fully fall into a predefined set of critical states that can characterize several system properties of interest. Petri nets and automata are the main tools for the modeling and control of DESs with potential mutual transformation [10].

There are two commonly employed techniques for checking the critical observability of a DES that is modeled with finite-state automata: regular language [11] and observer-based approaches [12–14]. In [11], the authors utilize the operations of regular languages

to develop a computational procedure to address the observability verification problem, executable in polynomial time. In [12], $k$-step critical observability is defined, and an observer called a $k$-extended detector is presented to verify $k$-step critical observability. The critical observability of large-scale finite-state automaton networks is considered in [13], and decentralized critical observers are designed. Lai et al. propose a novel method for constructing observers for polynomially ambiguous max-plus automata to check their critical observability, which extends earlier work on unambiguous max-plus automata [14]. More research on critical observability is displayed in Table 1.

**Table 1.** The current state of research on critical observability.

| Research Focus | System Model | Method | Contribution |
| --- | --- | --- | --- |
| Defining and verifying critical observability [15] | Petri nets | Using integer linear programming | The proposed method can be applied to Petri nets |
| Defining and verifying critical observability for networks of finite-state machines [13] | Finite-state machines | Constructing a decentralized key observer | A decentralized architecture is proposed for criticality observers of networks of finite-state machines |
| Checking critical observability [16] | Petri nets | Using integer linear programming | Addressing the problem of critical observability in the framework of labeled safe Petri nets |

When a system is verified to be non-critically observable, a significant problem is how to make it critically observable. This research reports a mechanism that, borrowing from DP [17], aims to consolidate/enforce the critical observability for a non-critically observable system. Further, the mechanism ensures that the critical states of the system cannot be detected by an intruder, i.e., achieving critical state protection/concealment.

Differential privacy (DP) is a data processing technique within the field of cryptography, first proposed by Dwork [17]. Building on a solid mathematical foundation, DP guarantees that the statistical characteristics exhibited by the overall dataset will almost remain unchanged when a single entity (a user, a record, or any form of individual data point) is added to or removed from the dataset. In essence, the inclusion/exclusion of a single record barely alters the computational outcomes of the dataset. Consequently, DP can withstand a wide range of privacy attacks, making it extremely difficult for attackers to accurately infer individual data by observing the calculation results.

The privacy protection mechanism due to DP is primarily achieved by introducing noise to obfuscate the original data during data processing, thereby concealing personal characteristics and sensitive information. The Laplace mechanism [18], which is particularly suited for protecting numerical data, and the exponential mechanism [19], designed for handling non-numerical data, are two of the most common and effective methods for injecting noise in order to implement DP.

As shown in the literature, DP has been widely applied in various fields, including big data analytics, federated learning, recommendation systems, and trajectory data publishing. In [20], DP is combined with the Haar wavelet technique to protect the privacy of big data in body sensor networks. The authors developed a tree-based structure to analyze the data, reducing errors and enabling long-range queries. In [21], the authors concentrated on three aspects of DP protection for adolescents' ultrasound examination health datasets: DP protection for basic statistical output perturbations, publication of DP marginal histograms and synthesized data, and machine learning DP learning algorithms. The problem of location and traffic information leakage in vehicular network environments is addressed in [22] by integrating local DP and federated learning. Specifically, the authors introduce four local DPMs, including the three-outputs and optimal piecewise mechanism, to perturb gradients from vehicles, ensuring privacy while maintaining model accuracy

and minimizing communication overhead. In [23], Ren et al. employ the data from various social network platforms to generate personalized recommendations where, to ensure the privacy of the users, DP is applied to the user attributes and the connections present in the social network graph. Presented in [24,25] is a privacy protection mechanism for symbolic systems whose traces (symbol sequences) can be protected. Based on this framework, the authors further develop an innovative exponential mechanism. In addition, DP finds its extensive applications in other domains such as logistics [26] and time series trajectories [27]. We believe that more research will be exposed as this kind of privacy protection paradigm prevails corresponding to the increasingly demanding requirements of system security.

Although DP is effective in protecting sensitive information within various information-intensive systems, its potential applications in DESs have yet to be fully explored. This paper primarily focuses on the problem of how to utilize DP to enforce critical observability in DESs modeled with finite-state automata. Furthermore, we explore how DP is employed to guarantee that the critical states are not discovered by an intruder. The research follows the schematic illustrated in Figure 1. Specifically, given a system $\mathcal{A}_{nd}$ modeled with a nondeterministic finite-state automaton, a set of observable events $\Sigma_o$, a set of critical states $X_s$, and a set of non-critical states $X_{ns}$, we assume that the system is not critically observable with respect to the predefined set of observable events $\Sigma_o$ and critical states $X_s$. Using the standard determinization procedure presented in [28], we construct a critical state estimator $Obs(\mathcal{A}_{nd})$ with respect to observable events. The system's behavior can be described as a scenario that the event sequence $s$ generated by $\mathcal{A}_{nd}$ is transmitted to the critical state estimator $Obs(\mathcal{A}_{nd})$, which then produces an observation $\rho$.

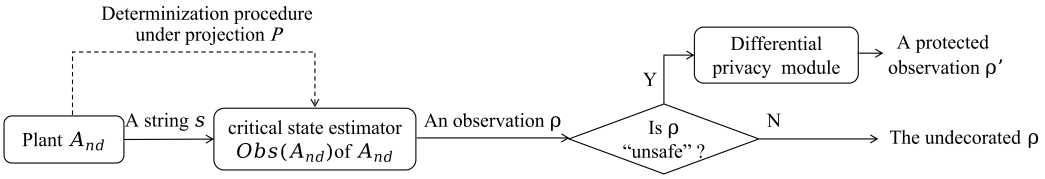



**Figure 1.** Framework of critical state concealment.

By estimating the state set that is consistent with an observation $\rho$, denoted by $\mathcal{C}(\rho)$, we draw a crucial conclusion. If this set is a subset of the non-critical states, i.e., $\mathcal{C}(\rho) \subseteq X_{ns}$, the observation $\rho$ is regarded as a safe observation. If the set is a subset of the critical states, i.e., $\mathcal{C}(\rho) \subseteq X_s$, the observation $\rho$ is considered as an unsafe observation. In both cases, the critical states can be revealed/detected by an intruder. Otherwise, the set $\mathcal{C}(\rho)$ is neither a subset of the critical states nor a subset of the non-critical states, i.e, $\mathcal{C}(\rho) \nsubseteq X_s \wedge \mathcal{C}(\rho) \nsubseteq X_{ns}$. In this situation, the observation $\rho$ is also deemed unsafe as the critical observability of the system is violated. In other words, when $\rho$ is acquired by an intruder, they can determine that the system is not critically observable. To prevent unsafe observations from being captured, a DP module is introduced as a protection shield. The function of this module is that, when an unsafe observation $\rho$ is received, the module outputs a protected observation $\rho'$ with a probability similar to that of another random observation, thereby obscuring the intruder such that it is difficult to trace back to the original unsafe observation. If the observation $\rho$ is safe, the system directly outputs the undecorated $\rho$.

We further analyze how DP works. For an unsafe observation $\rho$, it is first mapped into a state sequence $\omega$, which is a state string interleaved with $\rho$ in $Obs(\mathcal{A}_{nd})$, where $Obs(\mathcal{A}_{nd})$ is the observer of the plant $\mathcal{A}_{nd}$. Then, $\omega$ is sent to the DP module. We assume that the operational logic (running principals) of the DP module is public to an intruder, i.e., the intruder knows the probability distribution of the module generating all possible outputs $\rho'$ with respect to an input observation $\rho$. However, when the two state sequences, corresponding to an unsafe observation $\rho$ and a random observation $\tau$, respectively, are input into the module, the protected output observation $\rho'$ is highly similar in probability. This design guarantees that even if an intruder captures $\rho'$, it is difficult to distinguish whether it originates from $\rho$ or $\tau$, thereby effectively protecting unsafe observations that violate

critical observability (i.e., $\mathcal{C}(\rho) \nsubseteq X_s \wedge \mathcal{C}(\rho) \nsubseteq X_{ns}$) or reveal critical states (i.e., $\mathcal{C}(\rho) \subseteq X_s$). Under the aforementioned lines, the main contributions of this work are summarized as follows.

(1) A novel framework based on DP for the enforcement of critical observability for DESs is proposed. Given a non-critically observable system, we define an unsafe observation by computing the set of states consistent with it, i.e., an observational system output. In this paper, the designed method aims to confuse the unsafe observations with the randomly generated observations from a probabilistic viewpoint, such that a non-critically observable system satisfies critical observability. Moreover, this method guarantees that the critical states of the system cannot be detected by an intruder.
(2) To expand the application field of DP theory, we propose a new concept called "state sequence DP" and apply it to finite-state automata.
(3) Thanks to the concept of state sequence DP, we design an optimal DPM that ensures the optimal protection of unsafe observations.

The preliminaries of DESs and DP are presented in Section 2. The definition of critical observability is recalled in Section 3. In Section 4, a novel DPM specialized for the enforcement of critical observability of a system is designed. We discuss a realistic case of enforcing critical observability in Section 5 to show the practical applicability of the proposed approach. Finally, concluding remarks are given in Section 6.

## 2. Preliminaries

The basics of notations and notions necessary for the development of this research are presented, including languages and finite-state machines (deterministic finite automata—DFA; and non-deterministic finite automata—NFA), as they are appropriate for the modeling of a DES. Readers are referred to [28]. Note that an NFA is equivalent to a DFA from the lens of languages that they can generate, i.e, they have the same modeling power.

### *2.1. Languages and Automata*

Let $E$ be a finite set of event labels DES and $E^*$ be a set of the finite strings defined over $E$ (the Kleene closure of set $E$), consisting of the empty string $\varepsilon$. Given a string $\sigma \in E^*$, its length is the number of symbols contained in it, denoted by $|\sigma|$. In particular, we have $|\varepsilon| = 0$. Given two strings $\sigma_1$ and $\sigma_2$, the string $\sigma = \sigma_1\sigma_2$ is said to be the concatenation of $\sigma_1$ and $\sigma_2$.

A DFA is a quadruple $\mathcal{A}_d = (X_d, E, \delta_d, x_0)$, where $X_d$ is the finite set of states; $E$ is, as previously stated, the collection of event labels; $\delta_d : X_d \times E \to X_d$ is the partially defined deterministic transition function; and $x_0 \in X_d$ is the initial state. In order to facilitate the development of this research, one readily extends the transition function, as again denoted by $\delta_d$ in the case of no confusion, recursively defined from $X_d \times E \to X_d$ to $X_d \times E^* \to X_d$, where $\delta_d(x, se) = \delta_d(\delta_d(x, s), e)$ for $s \in E^*$ and $e \in E$ (specifically, we have $\delta_d(x, \varepsilon) = x$ for all $x \in X_d$). Let $k \in \mathbb{N}$ with $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ be a non-negative integer. We use $E^k$ to denote the set of event sequences with length $k$, i.e., $E^k = \{s \in E^* \mid |s| = k\}$. Analogously, we write $X_d^k$ to represent the set of all state sequences with length $k$. The language generated by $\mathcal{A}_d$ is defined as $\mathcal{L}(\mathcal{A}_d) = \{s \in E^* \mid \delta_d(x_0, s) \text{ is defined}\}$. The set of events that are enabled (feasible) at a given state $x \in X_d$ is defined by $\mathcal{E}(x) = \{e \in E \mid \delta_d(x, e) \text{ is defined}\}$. The number of enabled events at state $x$ is thus $|\mathcal{E}(x)|$.

Given a DFA $\mathcal{A}_d = (X_d, E, \delta_d, x_0)$, a sequence $\xi$ (interleaving of states and events) starting from state $x_0$ and ending at a state in $X_d$, i.e., $\xi = x_0 \xrightarrow{e_1} x_1 \xrightarrow{e_2} \cdots \xrightarrow{e_k} x_k$, is said to be a production of $\mathcal{A}_d$ with $e_i \in \mathcal{E}(x_{i-1})$ $(i = 1, 2, \ldots, k)$. We write $\Xi$ to represent the set of all productions $\xi$ in the DFA $\mathcal{A}_d$. Given a production $\xi = x_0 \xrightarrow{e_1} x_1 \xrightarrow{e_2} \cdots \xrightarrow{e_k} x_k$ of $\mathcal{A}_d$, the state sequence $x_0x_1 \cdots x_k$ associated with $\xi$ is called a production state sequence, denoted by $F(\xi) = x_0x_1 \cdots x_k$. Analogously, the event sequence $e_1e_2 \cdots e_k$ associated with $\xi$ is said to be a production event sequence, and it is denoted by $T(\xi) = e_1e_2 \cdots e_k$. Let $\gamma = x_0x_1 \cdots x_k$ with $k \geq 1$ be a non-empty state sequence. We write $\mathcal{T}(\gamma) = \{s \in E^* \backslash \{\varepsilon\} \mid \exists \xi \in \Xi : F(\xi) = \gamma \;\&\; s = T(\xi)\}$ to represent the set of event sequences consistent with $\gamma$.

Given a non-empty event sequence $s = e_1 \cdots e_k$ with $k \geq 1$, its consistent state sequence set is defined as $\mathcal{F}(s) = \{\gamma \in X_d^* \backslash \{\varepsilon\} \mid \exists \xi \in \Xi : T(\xi) = s \ \& \ \gamma = F(\xi)\}$. Note that $X_d^*$ is the set of all finite strings of elements in $X_d$.

**Example 1.** *Consider a DFA $\mathcal{A}_d = (X_d, E, \delta_d, x_0)$ shown in Figure 2, where $X_d = \{x_0, x_1, x_2, x_3, x_4\}$ and $E = \{\alpha, \beta, \lambda\}$. We have $\delta_d(x_0, \beta) = x_1$, $\delta_d(x_0, \lambda) = x_2$, $\delta_d(x_0, \alpha) = x_3$, $\mathcal{E}(x_0) = \{\beta, \lambda, \alpha\}$, and $|\mathcal{E}(x_0)| = 3$. Given a production $\xi = x_0 \xrightarrow{\alpha} x_3 \xrightarrow{\beta} x_4$, the production state sequence and the production event sequence associated with $\xi$ are $F(\xi) = x_0x_3x_4$ and $T(\xi) = \alpha\beta$, respectively. Let us consider an event sequence $\lambda\alpha$ in $\mathcal{A}_d$. The set of state sequences consistent with $\lambda\alpha$ is $\mathcal{F}(\lambda\alpha) = \{x_0x_2x_2\}$.*

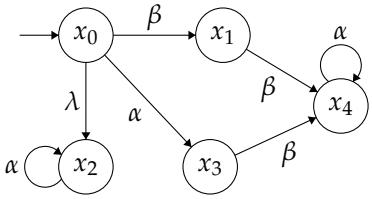



**Figure 2.** DFA $\mathcal{A}_d$.

An NFA is a quadruple $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$, where $X_{nd}$ is the finite set of states, $E$ is the same as explained before, $\delta_{nd} : X_{nd} \times E \to 2^{X_{nd}}$ is the (partial) transition function, and $X_0 \subseteq X_{nd}$ is the set of initial states. The transition function $\delta_{nd}$ is similarly extended to $\delta_{nd} : X_{nd} \times E^* \to 2^{X_{nd}}$ by defining $\delta_{nd}(x, \varepsilon) = \{x\}$ for all $x \in X_{nd}$ and $\delta_{nd}(x, se) = \bigcup_{x' \in \delta_{nd}(x,s)} \delta_{nd}(x', e)$, where $s \in E^*$ and $e \in E$. The language generated by $\mathcal{A}_{nd}$ from state $x \in X_{nd}$ is formally defined as $\mathcal{L}(\mathcal{A}_{nd}, x) = \{s \in E^* \mid \delta_{nd}(x, s) \neq \emptyset\}$. Given a set of states $X \subseteq X_{nd}$, we define $\mathcal{L}(\mathcal{A}_{nd}, X) = \cup_{x \in X} \mathcal{L}(\mathcal{A}_{nd}, x)$. The language of $\mathcal{A}_{nd}$ is $\mathcal{L}(\mathcal{A}_{nd}) = \mathcal{L}(\mathcal{A}_{nd}, X_0)$.

Given an automaton, $E$ is partitioned into two disjoint sets: $E = E_o \dot{\cup} E_{uo}$, where $E_o$ ($E_{uo}$) is the set of observable (unobservable) events. The natural projection $P : E^* \to E_o^*$ is defined by

$$P(\varepsilon) = \varepsilon \ \text{ and } \ P(se) = \begin{cases} P(s)e, & \text{if } e \in E_o; \\ P(s), & \text{if } e \in E_{uo}, \end{cases}$$

where $s \in E^*$ and $e \in E$. Function $P$ can be extended to $P : 2^{E^*} \to 2^{E_o^*}$, i.e., given $Z \subseteq E^*$, $P(Z) = \{z' \in E_o^* \mid \exists z \in Z : z' = P(z)\}$.

Given an NFA $\mathcal{A}_{nd} = (X_{nd}, E_o \dot{\cup} E_{uo}, \delta_{nd}, X_0)$ and a state set $X \subseteq X_{nd}$, the set of states unobservably reached from $X$ is defined as $UR(X) = \{x' \in X_{nd} \mid \exists x \in X, \exists s \in E_{uo}^* : x' \in \delta_{nd}(x, s)\}$. We use the notation $Ne(X, e)$ to represent the set of states reached by the occurrence of an observable event $e \in E_o$ from the states in $X$, i.e., $Ne(X, e) = \{x' \in X_{nd} \mid \exists x \in X : x' \in \delta_{nd}(x, e)\}$. The observer of $\mathcal{A}_{nd}$, denoted as $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$, can be computed following a standard procedure outlined in [28]. More specifically, $X_{obs} \subseteq 2^{X_{nd}}$ is the state set, $E_o$ is the set of observable events, $\delta_{obs} : X_{obs} \times E_o \to X_{obs}$ is the (partial) transition function (for all $x_{obs} \in X_{obs}$ and for all $e \in E_o$, $\delta_{obs}(x_{obs}, e) = UR(Ne(x_{obs}, e))$), and $x_{0,obs} = UR(X_0)$ is the initial state.

Due to the nature of partial observation of a considered plant, we can only observe the observable event labels, while unobservable events can be directly observed. However, different event sequences may have the same observational results. Given an NFA $\mathcal{A}_{nd}$, a string $\rho \in P(\mathcal{L}(\mathcal{A}_{nd}))$ is said to be an observation. The state estimation with respect to $\rho$ can be expressed as $\mathcal{C}(\rho) = \{x \in X_{nd} \mid \exists x_0 \in X_0, \exists s \in \mathcal{L}(\mathcal{A}_{nd}) : P(s) = \rho \ \& \ x \in \delta_{nd}(x_0, s)\}$. By definition, $\mathcal{C}(\rho) = \delta_{obs}(x_{0,obs}, \rho)$. Note that an observer is a DFA where all events are observable.

### 2.2. Differential Privacy

Differential privacy is an effective vehicle for privacy protection by providing a mechanism (or a function) to protect sensitive personal information from being disclosed during statistical analysis or data dissemination. Broadly speaking, whenever we use a dataset as the input for this mechanism, the addition or removal of any single record from the dataset will not affect the data query results. This means that even if the dataset is slightly modified, it is difficult or impossible for external intruders to obtain any specific privacy information [17].

**Definition 1** (Differential privacy [29])**.** *Let $\epsilon$ be a positive real number and $\mathcal{M}$ be a randomized mechanism (function) that takes a dataset as input. Let $Im(\mathcal{M})$ denote the image of $\mathcal{M}$. The mechanism $\mathcal{M}$ is said to provide $\epsilon$-differential privacy if, for any two datasets $D_1$ and $D_2$ that differ on a single element and for any $O \subseteq Im(\mathcal{M})$, it holds that*

$$\mathbb{P}[\mathcal{M}(D_1) \in O] \leq \exp(\epsilon)\mathbb{P}[\mathcal{M}(D_2) \in O],$$

*where the value of $\mathcal{M}$ at a dataset $D_1$ or $D_2$ is contained in the sample space, i.e., $Im(\mathcal{M})$, with a probability decided by the randomness used in the mechanism, and $\mathbb{P}[\mathcal{M}(D_1) \in O]$ is the probability of $\mathcal{M}(D_1) \in O$, representing the possibility that the output of $\mathcal{M}$ at $D_1$ belongs to $O$.*

In Definition 1, the performance of DP is evaluated by the value of $\epsilon$. Specifically, a smaller $\epsilon$ indicates a more subtle distinction between the probabilities of events $\mathcal{M}(D_1) \in O$ and $\mathcal{M}(D_2) \in O$, signifying a lower likelihood for an intruder to distinguish between the two datasets. Conversely, a larger value of $\epsilon$ implies a diminished level of protection for users' private information.

**Remark 1.** *According to Definition 1, two datasets, namely $D_1$ and $D_2$, are required to differ in only one element. However, in this study, we extend the concept of DP to the field of DESs modeled with a DFA $\mathcal{A}_d = (X_d, E, \delta_d, x_0)$ (the formal definition of DP within the DFAs framework, called state sequence DP, will be presented in Section 4). In the definition of DP tailored for DESs, we define $\mathcal{M}_\omega : X_d^k \to X_d^k$ as a mechanism specifically designed for a set of state sequences. The input and output of $\mathcal{M}_\omega$ are two state sequences of equal length. The degree of similarity between these sequences is assessed based on their Hamming distance, as defined in Section 4.*

## 3. Critical Observability for Automata

In this section, we recall the formalization of critical observability verification problems. Given an NFA $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$ that is partially observable, i.e., $E = E_o \dot{\cup} E_{uo}$, in practice, certain important physical states of the NFA are called critical states, represented by a subset of states $X_s \subseteq X_{nd}$. By acquiring a random sequence of observable events, if an intruder can always unambiguously infer whether the system is currently in a predefined set of critical states, $\mathcal{A}_{nd}$ is said to be critically observable.

**Definition 2** (Critical observability [13])**.** *Given an NFA $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$, a set of observable events $E_o \subseteq E$, and a set of critical states $X_s \subseteq X_{nd}$, the system $\mathcal{A}_{nd}$ is said to be critically observable with respect to $E_o$ and $X_s$ if the following predicate holds*

$$\forall \rho \in P(\mathcal{L}(\mathcal{A}_{nd})) : [\mathcal{C}(\rho) \subseteq X_s] \vee [\mathcal{C}(\rho) \subseteq X_{nd} \backslash X_s].$$

A critically observable system requires that, for any observation $\rho \in P(\mathcal{L}(\mathcal{A}_{nd}))$, the state set consistent with the observation should be either a subset of critical states or a subset of non-critical states. To verify whether an NFA $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$ is critically observable, one can construct a critical state estimator of $\mathcal{A}_{nd}$, i.e., the observer $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ of $\mathcal{A}_{nd}$, which stores all possible state estimations [28]. Then, a system $\mathcal{A}_{nd}$ is critically observable with respect to $X_s$ and $E_o$ if for any state $x' \in X_{obs}$, either $x' \subseteq X_s$ or $x' \subseteq X_{nd} \backslash X_s$ holds.

Let $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$ be an NFA that is not critically observable, $E_o \subseteq E$ be a set of observable events, $X_s \subseteq X_{nd}$ be a set of critical states, and $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ be the critical state estimator of $\mathcal{A}_{nd}$. The set of critical observability violative states is denoted as $X_N = \{x \in X_{obs} \mid x \nsubseteq X_s \wedge x \nsubseteq X_{nd} \backslash X_s\}$.

**Example 2.** *Consider the NFA $\mathcal{A}_{nd} = \{X_{nd}, E, \delta_{nd}, X_0\}$ shown in Figure 3, where $X_{nd} = \{x_0, x_1, x_2, x_3, x_4, x_5\}$ is the set of states, $E = E_o \cup E_{uo} = \{\alpha, \beta, \theta, \lambda\}$ is the set of events with $E_o = \{\alpha, \beta, \lambda\}$ and $E_{uo} = \{\theta\}$, and $X_0 = \{x_0\}$ is the set of initial states. The critical state estimator $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ of $\mathcal{A}_{nd}$ is portrayed in Figure 4. We assume that the set of critical states is $X_s = \{x_3, x_4\}$. In Figure 4, by considering an observation $\rho = \beta\alpha\lambda$, the system $\mathcal{A}_{nd}$ is not critically observable with respect to $E_o$ and $X_s$, thanks to $\mathcal{C}(\beta\alpha\lambda) = \delta_{obs}(x_{0,obs}, \beta\alpha\lambda) = x_{o4} = \{x_3, x_4, x_5\}$, i.e, there exist a state $x_{o4} \in X_{obs}$ such that $x_{o4} \nsubseteq X_s$ and $x_{o4} \nsubseteq X_{nd} \backslash X_s$.*

*However, the NAF $\mathcal{A}_{nd}$ is critically observable if the critical state is $X_s = \{x_2\}$, as intruders can always determine whether the critical state is reached by observing the system's behavior. That is, in $Obs(\mathcal{A}_{nd})$, for all $x \in X_{obs}$, $x \subseteq X_s$ or $x \subseteq X_{nd} \backslash X_s$ holds.*

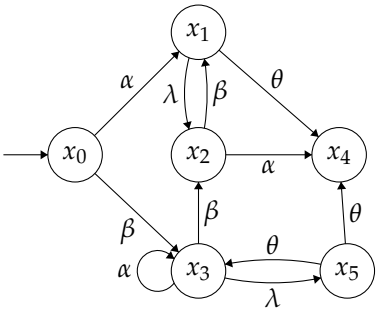
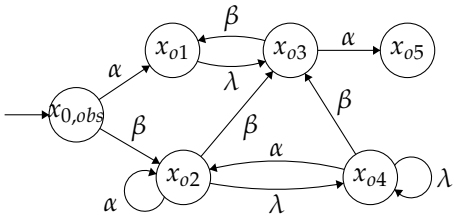



**Figure 3.** NFA $\mathcal{A}_{nd}$.




$x_{0,obs} = \{x_0\}$ $x_{o1} = \{x_1, x_4\}$ $x_{o2} = \{x_3\}$
$x_{o3} = \{x_2\}$ $x_{o4} = \{x_3, x_4, x_5\}$ $x_{o5} = \{x_4\}$

**Figure 4.** Critical state estimator $Obs(\mathcal{A}_{nd})$ of $\mathcal{A}_{nd}$.

Consider a system $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$ that is not critically observable with respect to the set of observable events $E_o \subseteq E$ and the set of critical states $X_s \subseteq X_{nd}$. Let $Obs(\mathcal{A}_{nd})$ be the critical state estimator of $\mathcal{A}_{nd}$. Given a production $\xi = x_0 \xrightarrow{e_1} x_1 \xrightarrow{e_2} \cdots \xrightarrow{e_k} x_k$ of $Obs(\mathcal{A}_{nd})$ with the observation $\rho = e_1 e_2 \cdots e_k$, if the set of states consistent with $\rho$ is a subset of critical observability violative states or a subset of critical states, i.e., $\mathcal{C}(\rho) \subseteq X_N$ or $\mathcal{C}(\rho) \subseteq X_s$, the production is said to be unsafe. In this case, the corresponding sequences $F(\xi) = x_0 x_1 \cdots x_k$ and $T(\xi) = \rho = e_1 e_2 \cdots e_k$ are defined as an unsafe production state sequence and an unsafe production event sequence (or an unsafe observation), respectively. Otherwise, if the set of states consistent with $\rho$ is included in $X_{nd} \backslash X_s$, i.e., $\mathcal{C}(\rho) \subseteq X_{nd} \backslash X_s$, the production is identified as safe. Its corresponding production event sequence $T(\xi)$ and production state sequence $F(\xi)$ are also considered safe. Once an unsafe observation is exposed to an intruder, they can determine that the system is not critically observable or the system's critical states are revealed. In the following, we focus on how to make a non-critically observable system satisfy critical observability. Moreover, we guarantee that the critical states of the system cannot be revealed by an intruder.

## 4. Critical State Concealment Based on Differential Privacy

In this section, we aim to design a DPM to enforce critical observability for DESs modeled with DFAs and to ensure that the predefined critical states cannot be detected to intruders. Specifically, we first define the Hamming distance of two state sequences in a DFA to describe the difference between them. Then, to prevent critical states or critical observability violative states from being captured by an intruder, we define a novel DPM for DFAs, called state sequence DP. Finally, an algorithm that provides a probability distribution for the generation of each state in an output state sequence is proposed, and the DPM is designed.

Currently, the existing methods for enforcing critical observability or opacity are mainly based on supervisory control theory [30–32] and edit functions [33–35]. Supervisory control theory ensures the critical observability or opacity of the system by constructing a supervisor. The edit function prevents the disclosure of critical or secret states by changing the length of the observations (deletion or insertion events). The difference between the method proposed in this paper and these available so far is that, from a probabilistic viewpoint, we confuse unsafe observations with randomly generated observations such that a non-critically observable system satisfies critical observability, and the proposed method does not change the length of the observations.

### *4.1. Problem Statement*

In $Obs(\mathcal{A}_{nd})$, when an unsafe observation $\rho$ is generated, we can design a DPM associated with $\rho$ to effectively conceal the unsafe observation. In particular, given an observation $\rho$, we derive its corresponding production state sequence $\omega_1 \in \mathcal{F}(\rho)$. By inputting $\omega_1$ and a random production state sequence $\omega_2$, such that the Hamming distance between $\omega_1$ and $\omega_2$ is restricted to a bound $l \in \mathbb{N}$ into the designed mechanism, a protected production state sequence $\omega_3$ is output with approximate probability, where the lengths of $\omega_1$, $\omega_2$, and $\omega_3$ are equal (i.e., $|\omega_1| = |\omega_2| = |\omega_3|$). Because both $\omega_1$ and $\omega_2$ can lead to the generation of $\omega_3$ with probabilities that are sufficiently close, it becomes impossible for an intruder to determine whether $\omega_1$ or $\omega_2$ is generated by the estimator $Obs(\mathcal{A}_{nd})$. In other words, from the observed behavior contained in $\mathcal{T}(\omega_3)$, the intruder cannot infer its corresponding input behavior, and therefore the unsafe observation is not identifiable. Before discussing the use of DP to protect unsafe observations from being acquired, we first provide some definitions.

**Definition 3** (Hamming distance [36]). *Given two production state sequences $\omega_1 = x_0x_1\cdots x_k$ and $\omega_2 = x'_0x'_1\cdots x'_k$ generated by a critical state estimator $Obs(\mathcal{A}_{nd})$ with the same length ($|\omega_1| = |\omega_2|$), the Hamming distance between $\omega_1$ and $\omega_2$ is defined as $\mathcal{J}(\omega_1,\omega_2) = |\{j \mid \omega_1(j) \neq \omega_2(j)\}|$, where $\omega_i(j)$ represents the j-th entry of production state sequence $\omega_i$, $i = 1,2$, and $j = 1,2,\ldots,|\omega_1|$.*

Let $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ be a critical state estimator of $\mathcal{A}_{nd}$, $k \in \mathbb{N}^+ = \{1,2,\ldots\}$ be a positive integer representing the length of a production state sequence, and $l \in \mathbb{N}^+$ be a parameter. We define $\mathcal{W}_{k,l}$ as the set of all pairs of production state sequences $(\omega_1,\omega_2)$ with $|\omega_1| = |\omega_2| = k$ and the Hamming distance $\mathcal{J}(\omega_1,\omega_2)$ between them being less than or equal to $l$, i.e.,

$$\mathcal{W}_{k,l} = \{(\omega_1,\omega_2) \in X^k_{obs} \times X^k_{obs} \mid \mathcal{J}(\omega_1,\omega_2) \leq l\}. \tag{1}$$

Considering Definition 1 and the properties of DES models, we now define state sequence DP in a DFA.

**Definition 4** (State sequence differential privacy [37]). *Given an NFA $\mathcal{A}_{nd}$, let $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ be its critical state estimator and $X^k_{obs}$ be the set of state sequences with length k in $Obs(\mathcal{A}_{nd})$. Let $\epsilon \geq 0$ be a real number. A mechanism $\mathcal{M}_\omega : X^k_{obs} \to X^k_{obs}$ is said to be of state*

*sequence DP with respect to $\mathcal{W}_{k,l}$ if for all production state sequence pairs $(\omega_1, \omega_2) \in \mathcal{W}_{k,l}$ and for all $\omega_3 \in X_{obs}^k$, it holds that*

$$\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3] \leq \exp(\epsilon)\mathbb{P}[\mathcal{M}_\omega(\omega_2) = \omega_3], \tag{2}$$

*where $\mathbb{P}[\mathcal{M}_\omega(\omega_i) = \omega_3]$ represents the probability of $\mathcal{M}_\omega(\omega_i) = \omega_3$ with $i \in \{1, 2\}$.*

According to Definition 4, a mechanism that guarantees DP of state sequences with respect to $\mathcal{W}_{k,l}$ implies that for all pairs of production state sequences $(\omega_1, \omega_2) \in \mathcal{W}_{k,l}$ with $\omega_1, \omega_2 \in X_{obs}^k$, and for all possible output production state sequences $\omega_3 \in X_{obs}^k$, the probability of $\omega_1$ being decorated as $\omega_3$ and that of $\omega_2$ being decorated as $\omega_3$ is close enough, such that an intruder cannot identify which is the real input to $\mathcal{M}_\omega$.

To evaluate the performance of the mechanism, we introduce the concept of the security degree of a state sequence. Given an NFA $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$, a set of observable events $E_o \subseteq E$, a set of critical states $X_s \subseteq X_{nd}$, and the critical state estimator $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ of $\mathcal{A}_{nd}$, let $\omega = x_0 x_1 \cdots x_k \in X_{obs}^{k+1}$ be a state sequence in $Obs(\mathcal{A}_{nd})$. If the production state sequence $\omega$ is safe, the security degree of $\omega$ is defined as $deg(\omega) = |x_k|$, where $x_k = \mathcal{C}(\rho)$ with $\rho \in \mathcal{T}(\omega)$. Otherwise, the sequence $\omega$ is unsafe, whose security degree is defined as $deg(\omega) = 0$. In other words, the state sequence $\omega$ consistent with an observation $\rho$ that violates critical observability or reveals the critical states (i.e., $\mathcal{C}(\rho) \subseteq X_N$ or $\mathcal{C}(\rho) \subseteq X_s$) has the lowest security level. Note that for all $\rho \in \mathcal{T}(\omega)$, $\mathcal{C}(\rho)$ is unique. Given a production state sequence $\omega$ in $Obs(\mathcal{A}_{nd})$, a larger value $deg(\omega)$ means that the set of critical observability violative states $X_N$ or that the set of critical states $X_s$ is less likely to be detected by an intruder.

Consider a critical state estimator $Obs(\mathcal{A}_{nd})$ of a system $\mathcal{A}_{nd}$ and a parameter $k \in \mathbb{N}^+$. The maximum security degree of production state sequences $\omega = x_0 x_1 \cdots x_k \in X_{obs}^{k+1}$ contained in $Obs((\mathcal{A}_{nd})$ is defined as $deg_{max}(Obs(\mathcal{A}_{nd}), k+1) = max\{g \in \mathbb{N} \mid \exists \omega \in X_{obs}^{k+1} : deg(\omega) = g\}$ (the notation *max* (or *min*) applied to a set represents the maximum (or minimum) element contained in the set).

**Example 3.** *Consider the NFA $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$ and a set of critical states $X_s = \{x_3\}$ in Example 2. Let $\xi = x_{0,obs} \xrightarrow{\alpha} x_{o1} \xrightarrow{\lambda} x_{o3} \xrightarrow{\beta} x_{o1}$ be a production of $Obs(\mathcal{A}_{nd})$ portrayed in Figure 4 with $F(\xi) = \omega = x_{0,obs} x_{o1} x_{o3} x_{o1}$. The security degree of $\omega$ is 2, i.e., $deg(\omega) = |\{x_1, x_4\}| = 2$. Given another production $\xi' = x_{0,obs} \xrightarrow{\beta} x_{o2} \xrightarrow{\lambda} x_{o4}$, the production state sequence $F(\xi') = \omega' = x_{0,obs} x_{o2} x_{o4}$ is unsafe, and $\omega'$ has the lowest security degree $deg(\omega') = 0$ thanks to $x_{o4} \subseteq X_N$. Given a non-negative integer $k = 4$ representing the length of a state sequence, we have $deg_{max}(Obs(\mathcal{A}_{nd}), 4) = 2$.*

In this paper, we focus on a mechanism that provides the highest degree of protection for an unsafe production state sequence. In simpler terms, the mechanism is capable of confusing unsafe state sequences with safe ones that have the maximum security degree. In particular, the designed mechanism with respect to $\mathcal{W}_{k+1,l}$ ensures that, for an unsafe production state sequence $\omega_1 = x_0 x_1 \cdots x_k$, there exists a production state sequence $\omega_2 = x'_0 x'_1 \cdots x'_k$ with maximum degree $deg_{max}(Obs(\mathcal{A}_{nd}), k+1)$ such that $(\omega_1, \omega_2) \in \mathcal{W}_{k+1,l}$ and $\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3] \leq \exp(\epsilon)\mathbb{P}[\mathcal{M}_\omega(\omega_2) = \omega_3]$ for all $\omega_3 \in X_{obs}^{k+1}$.

The distance bound $l$ is another parameter that significantly impacts the performance of the mechanism $\mathcal{M}_\omega$. Given an unsafe state sequence $\omega_1$, it is crucial to select a value $l$ that is as small as possible while guaranteeing that the mechanism $\mathcal{M}_\omega$ can maximally confuse $\omega_1$ with $\omega_2$. In other words, the state sequence $\omega_2$, which serves as a suitable alternative for $\omega_1$, should exhibit the utmost similarity to $\omega_1$.

**Definition 5** ([37]). *Let $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$ be an NFA that is not critically observable, $X_s \subseteq X_{nd}$ be a set of critical states, $E_o \subseteq E$ be a set of observable events, $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ be a critical state estimator of $\mathcal{A}_{nd}$, and $\mathcal{M}_\omega : X_{obs}^{k+1} \to X_{obs}^{k+1}$ be a mechanism*

*that provides state sequence DP with respect to $\mathcal{W}_{k+1,l}$. Given an unsafe state sequence $\omega_1$, the mechanism $\mathcal{M}_\omega$ is optimal with respect to $\omega_1$ if it satisfies the following statements:*

1. *There exists a production state sequence $\omega_2$ in $Obs(\mathcal{A}_{nd})$ with $deg(\omega_2) = deg_{max}$ $(Obs(\mathcal{A}_{nd}), k+1)$ such that $(\omega_1, \omega_2) \in \mathcal{W}_{k+1,l}$ holds;*
2. *For all $\omega_3 \in X_{obs}^{k+1}$, the inequality $\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3] \leq \exp(\epsilon)\mathbb{P}[\mathcal{M}_\omega(\omega_2) = \omega_3]$ holds;*
3. *For all $\omega_2$ in $Obs(\mathcal{A}_{nd})$ with $deg(\omega_2) = deg_{max}(Obs(\mathcal{A}_{nd}), k+1)$, there does not exist $l' < l$ $(l, l' \in \mathbb{N})$ such that $\mathcal{J}(\omega_1, \omega_2) = l'$.*

According to Definition 5, given an unsafe $\omega_1$, Condition (1) ensures that there exists at least one state sequence $\omega_2$ with the maximum security degree such that $(\omega_1, \omega_2)$ belongs to the set $\mathcal{W}_{k+1,l}$. Condition (2) indicates that an optimal mechanism provides (guarantees) the state sequence DP with respect to $\mathcal{W}_{k+1,l}$. Condition (3) implies that the parameter $l$ with $l = \mathcal{J}(\omega_1, \omega_2)$ must be minimal for all state sequences $\omega_2$ that satisfy $deg(\omega_2) = deg_{max}(Obs(\mathcal{A}_{nd}), k+1)$. In the following, we will detail the design of an optimal mechanism $\mathcal{M}_\omega$ that satisfies the definition of state sequence DP.

### *4.2. State Sequence Differential Privacy*

Based on Definitions 4 and 5, we first design a probability allocation strategy for the optimal mechanism $\mathcal{M}_\omega$ in a critical state estimator $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ of system $\mathcal{A}_{nd}$. Given an unsafe input production state sequence $\omega_1 = x_0 x_1 \cdots x_k \in X_{obs}^{k+1}$, the aim is to generate a random state sequence $\omega_3 = x_0'' x_1'' \cdots x_k'' \in X_{obs}^{k+1}$ with a certain probability from $\mathcal{M}_\omega$ such that $\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3]$. Specifically, the strategy to be defined enables assigning a probability $\mathbb{P}[x_i'']$ to each state included in the output production state sequence $\omega_3 = x_0'' x_1'' \cdots x_k'' \in X_{obs}^{k+1}$, where $i = 0, \ldots, k$. Then, by multiplying the probabilities $\mathbb{P}[x_i'']$ of all output states $x_i'' \in X_{obs}$ $(i = 0, 1, \ldots, k)$, we can obtain the probability of generating $\omega_3$, i.e., $\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3] = \mathbb{P}[x_0''] \times \mathbb{P}[x_1''] \times \cdots \times \mathbb{P}[x_k'']$. In terms of the strategy, we explore the conditions for ensuring the DP of state sequences and further design an algorithm to synthesize this optimal mechanism $\mathcal{M}_\omega$.

Given a critical state estimator $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$, let $\mathcal{M}_\omega : X_{obs}^{k+1} \to X_{obs}^{k+1}$ be a mechanism that takes a production state sequence $\omega_1 = x_0 x_1 \cdots x_k \in X_{obs}^{k+1}$ as the input and $\omega_3 = x_0'' x_1'' \cdots x_k'' \in X_{obs}^{k+1}$ as the output. Before elaborating on the probability allocation strategy to a state in $\omega_3$, we introduce an indicator function denoted as $\nu(x_{i-1}'', x_i)$ for two states $x_{i-1}'', x_i \in X_{obs}$, defined as follows:

$$\nu(x_{i-1}'', x_i) = \begin{cases} 1, & \text{if } \exists e \in E_o : \delta_{obs}(x_{i-1}'', e) = x_i; \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $i = 1, 2, \cdots, k$. To put it simply, in $Obs(\mathcal{A}_{nd})$, if there exists an observable event $e$ such that the transition function $\delta_{obs}(x_{i-1}'', e) = x_i$ holds, then the value of the indicator function $\nu(x_{i-1}'', x_i)$ is 1; otherwise $\nu(x_{i-1}'', x_i) = 0$. If $x_i''$ included in $\omega_3$ satisfies $x_i'' = x_i$, it is considered a correct state. On the other hand, if $x_i'' \neq x_i$, the state $x_i''$ is referred to as an incorrect state.

Now, we describe a probability allocation strategy that assigns a probability $\mathbb{P}[x_i'']$ to each state $x_i''$ in the state sequence $\omega_3 = x_0'' x_1'' \cdots x_k'' \in X_{obs}^{k+1}$, where $i \in \{0, \ldots, k\}$. Given an input production state sequence $\omega_1 = x_0 x_1 \cdots x_k \in X_{obs}^{k+1}$, let $\mathcal{X}_{\omega_1}$ be the set that collects all the states in $\omega_1$, i.e., $\mathcal{X}_{\omega_1} = \{x_0, x_1, \ldots, x_k\}$. The strategy for $\omega_3$ associated with $\omega_1$ is a (partial) function $\varphi : X_{obs} \times \mathcal{X}_{\omega_1} \times X_{obs} \to [0, 1]$ that outputs a probability of generating state $x_i'' \in X_{obs}$ in relation to two states $x_{i-1}'' \in X_{obs}$ and $x_i \in \mathcal{X}_{\omega_1}$, where $(x_{i-1}'', x_i, x_i'') \in X_{obs} \times \mathcal{X}_{\omega_1} \times X_{obs}$. Particularly, for the initial state, we define $x_0 = x_0'' = x_{0,obs}$, and it is assigned a probability value $\mathbb{P}[x_0''] = 1$. Then, for all $i \in \{1, \ldots, k\}$, if $x_i''$ is a correct state, it is allocated a correct probability $\varphi(x_{i-1}'', x_i, x_i'') = \mathbb{P}[x_i''] = \eta(x_i'')$ (the value of $\eta(x_i'')$ is examined in detail in Theorem 1). On the other hand, if $x_i''$ is not a correct state, let

$$\varphi((x''_{i-1}, x_i, x''_i) = \mathbb{P}[x''_i] = \frac{1 - \eta(x''_i)\nu(x''_{i-1}, x_i)}{\left|\mathcal{E}(x''_{i-1})\right| - \nu(x''_{i-1}, x_i)}. \tag{4}$$

Given an input production state sequence $\omega_1$, a randomized production state sequence is generated by using the probability allocation strategy $\varphi$. We use $\Omega$ to denote the set of all possible outputs, i.e., $\Omega = \{\omega_3^1, \omega_3^2, \cdots, \omega_3^n\}$, where $n \in \mathbb{N}^+$, $n$ represents the number of possible outputs. The strategy $\varphi$ ensures that $\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3^1] + \mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3^2] + \cdots + \mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3^n] = 1$, suggesting that the sum of probabilities for all outputs is equal to 1.

Then, we design an algorithm (namely Algorithm 1) to compute the probability allocation strategy $\varphi$ for a random output $\omega_3 = x''_0 x''_1 \cdots x''_k$ associated with an input state sequence $\omega_1 = x_0 x_1 \cdots x_k$. The algorithm takes as the input a critical state estimator, an input state sequence, probabilities of correct states, and an indicator function. It iterates through each state in the input sequence and calculates the probability of generating each possible output state based on the given probabilities of correct states and the indicator function. In Algorithm 1, for all $i = 1, \ldots, k$, if $i = 1$, we have $x''_0 = x_0 = x_{0,obs}$. For each possible state $x''_i \in X_{obs}$, it checks whether $x''_i = x_i$ and whether the indicator function $\nu(x''_{i-1}, x''_i) = 1$. If both conditions hold, the correct state $x''_i$ is allocated a correct probability $\eta(x''_i)$. If $x''_i \neq x_i$ and the indicator function $\nu(x''_{i-1}, x''_i) = 1$, the state $x''_i$ is assigned an incorrect probability

$$\frac{1 - \eta(x''_i)\nu(x''_{i-1}, x_i)}{\left|\mathcal{E}(x''_{i-1})\right| - \nu(x''_{i-1}, x_i)}. \tag{5}$$

Otherwise, if $\nu(x''_{i-1}, x''_i) = 0$, $\varphi(x''_{i-1}, x_i, x''_i) = 0$ is defined. If $1 < i \leq k$, for all $x''_{i-1} \in X_{obs}$, the codes in lines 4–10 are repeated. In simple terms, for each output state $x''_i$, where $i \in \{1, ..., k\}$, its probability value is associated with two states, i.e., $x''_{i-1}$ and $x_i$. If the output state $x''_i$ is feasible from $x''_{i-1}$ (i.e., $\nu(x''_{i-1}, x''_i) = 1$) and $x''_i$ is the same as its corresponding input state $x_i$ (i.e., $x''_i = x_i$), then the output state $x''_i$ is a correct state and is assigned a correct probability $\eta(x''_i)$. If the output state $x''_i$ starting from $x''_{i-1}$ is feasible but the state $x''_i$ is different from its corresponding input state $x_i$, then these output states are incorrect states and are assigned the incorrect probability value $\frac{1-\eta(x''_i)\nu(x''_{i-1}, x_i)}{\left|\mathcal{E}(x''_{i-1})\right| - \nu(x''_{i-1}, x_i)}$. The computational complexity of Algorithm 1 is $\mathcal{O}(k \times |X_{obs}|)$.

**Example 4.** *Consider a state sequence $\omega_1 = x_0 x_1 x_2 = x_{0,obs} x_{o2} x_{o4}$ in the critical state estimator in Figure 4 and feed into a mechanism $\mathcal{M}_\omega$. Let us elaborate on how to calculate the probability allocation strategy $\varphi$ for a randomly generated output state sequence $\omega_3 = x''_0 x''_1 x''_2$ with $x''_0 = x_0 = x_{0,obs}$ (all possible output state sequences $\omega_3$ with $|\omega_3| = 3$ are illustrated in Figure 5).*

*If $x''_1 = x_1 = x_{o2}$, according to the strategy $\varphi$, the state $x''_1$ is correct and is assigned a correct probability $\eta(x''_1)$ with $\varphi(x''_0, x_1, x''_1) = \varphi(x_{0,obs}, x_{o2}, x_{o2}) = \eta(x''_1)$ (the value of $\eta(x''_1)$ is discussed in Theorem 1); if $x''_1 = x_{o1}$ with $x_{o1} \neq x_{o2}$, by line 8 of Algorithm 1, we allocate an incorrect probability to state $x_{o1}$, i.e., $\varphi(x''_0, x_1, x''_1) = \varphi(x_{0,obs}, x_{o2}, x_{o1}) = \frac{1-\eta(x''_1)\times 1}{2-1} = 1 - \eta(x''_1)$.*

*At the state $x''_1 = x_{o1}$, as the feasible state $x''_2 = x_{o3}$ is not correct, i.e., $x''_2 \neq x_2$ and $\nu(x''_1, x''_2) = \nu(x_{o1}, x_{o3}) = 1$, we have $\varphi(x''_1, x_2, x''_2) = \varphi(x_{o1}, x_{o4}, x_{o3}) = \frac{1-\eta(x''_1)\times 0}{1-0} = 1$. Analogously, at the state $x''_1 = x_{o2}$, we have $\varphi(x_{o2}, x_{o4}, x_{o4}) = \eta(x''_2)$ and $\varphi(x_{o2}, x_{o4}, x_{o2}) = \varphi(x_{o2}, x_{o4}, x_{o3}) = \frac{1-\eta(x''_2)\times 1}{3-1} = \frac{1-\eta(x''_2)}{2}$.*

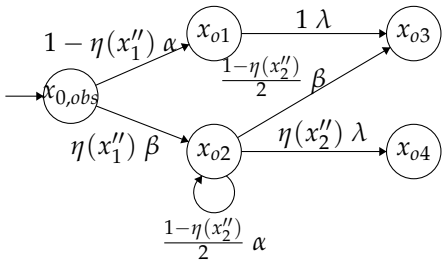



**Figure 5.** Probability allocation strategy for all possible output production state sequences $\omega_3 = x_0''x_1''x_2'' \in X_{obs}^3$.

**Algorithm 1:** A probability allocation strategy $\varphi$.

**Input:** A critical state estimator $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$, an input state sequence $\omega_1 = x_0x_1 \ldots x_k$ contained in $Obs(\mathcal{A}_{nd})$, probabilities of correct states $\eta(x_i'')$ with $i = 1, 2, \ldots, k$, and an indicator function $\nu$.

**Output:** Probability allocation strategy $\varphi$ for an output $\omega_3 = x_0''x_1'' \cdots x_k''$.

1 **for all** $i = 1$ to $k$ **do**
2  **if** $i = 1$ **then**
3  $x_{i-1}'' := x_{i-1} := x_{0,obs}$;
4  **for all** $x_i'' \in X_{obs}$ **do**
5  **if** $x_i'' = x_i$ and $\nu(x_{i-1}'', x_i'') = 1$ **then**
6  $\varphi(x_{i-1}'', x_i, x_i'') := \eta(x_i'')$;
7  **else if** $x_i'' \neq x_i$ and $\nu(x_{i-1}'', x_i'') = 1$ **then**
8  $\varphi(x_{i-1}'', x_i, x_i'') := \frac{1-\eta(x_i'')\nu(x_{i-1}'', x_i)}{|\mathcal{E}(x_{i-1}'')|-\nu(x_{i-1}'', x_i)}$;
9  **else**
10  $\varphi(x_{i-1}'', x_i, x_i'') := 0$;
11  **else**
12  **for all** $x_{i-1}'' \in X_{obs}$ **do**
13  Repeat the codes in lines 4–10;

By utilizing the probability allocation strategy $\varphi$, we are able to synthesize a mechanism that generates a production state sequence assigned with a probability. In other words, for all states contained in $\omega_3 = x_0''x_1'' \cdots x_k''$, the probability of generating $\omega_3$ is given by $\prod_{i=1}^{k} \varphi(x_{i-1}'', x_i, x_i'')$ with $\mathbb{P}[x_0''] = \mathbb{P}[x_{0,obs}] = 1$. Now, we demonstrate the condition of $\eta(x_i'')$ for a mechanism $\mathcal{M}_\omega$ to provide state sequence DP with respect to $\mathcal{W}_{k+1,l}$, based on Algorithm 1.

**Theorem 1.** *Given a critical state estimator $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ of a system $\mathcal{A}_{nd}$ that is not critically observable, let $\epsilon > 0$ be a real number, $l \in \mathbb{N}$ be a parameter, $k \in \mathbb{N}^+$ be the length of a state sequence, $\mathcal{M}_\omega : X_{obs}^{k+1} \to X_{obs}^{k+1}$ be a mechanism whose probability allocation strategy is computed by Algorithm 1, and $x_i''$ $(0 \leq i \leq k)$ be a correct state contained in an output state sequence $\omega_3 = x_0''x_1'' \cdots x_k'' \in X_{obs}^{k+1}$ of $\mathcal{M}_\omega$. The mechanism $\mathcal{M}_\omega$ guarantees state sequence DP with respect to $\mathcal{W}_{k+1,l}$ if the probabilities of correct states $\eta(x_i'')$ satisfy the following condition:*

$$\eta(x_i'') = \frac{1}{(|\mathcal{E}(x_{i-1}'')| - 1)\exp(-\frac{\epsilon}{l}) + 1}. \tag{6}$$

**Proof.** Let $\omega_1 = x_0x_1 \cdots x_k$ and $\omega_2 = x_0'x_1' \cdots x_k'$ be two production state sequences generated by $Obs(\mathcal{A}_{nd})$ such that $(\omega_1, \omega_2) \in \mathcal{W}_{k+1,l}$. We consider $\mathcal{M}_\omega(\omega_1) = \mathcal{M}_\omega(\omega_2) = \omega_3$ with $\omega_3 = x_0''x_1'' \cdots x_k'' \in X_{obs}^{k+1}$ (the initial states in $\omega_1$, $\omega_2$, and $\omega_3$ are the same, i.e., $x_0 = x_0' = x_0'' = x_{0,obs}$). By Definition 4, we need to prove the following inequality:

$$\exp(-\epsilon) \leq \frac{\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3]}{\mathbb{P}[\mathcal{M}_\omega(\omega_2) = \omega_3]} \leq \exp(\epsilon).$$

To begin with, we show $\frac{\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3]}{\mathbb{P}[\mathcal{M}_\omega(\omega_2) = \omega_3]} \leq \exp(\epsilon)$. Using Algorithm 1, we have

$$\begin{aligned}\frac{\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3]}{\mathbb{P}[\mathcal{M}_\omega(\omega_2) = \omega_3]} &= \frac{\eta(x_0'') \times \prod_{i=1}^{k} \varphi(x_{i-1}'', x_i, x_i'')}{\eta(x_0'') \times \prod_{j=1}^{k} \varphi(x_{j-1}'', x_j', x_j'')} \\ &= \prod_{r \in R} \frac{\varphi(x_{r-1}'', x_r, x_r'')}{\varphi(x_{r-1}'', x_r', x_r'')},\end{aligned}$$

where $R = \{r \in \mathbb{N}^+ \mid x_r \neq x_r'\}$ and $|R| \leq l$.

Note that when $x_i''$ is an incorrect state, the values of $\varphi(x_{i-1}'', x_i, x_i'')$ depend on $\nu(x_{i-1}'', x_i)$, i.e.,

$$\varphi(x_{i-1}'', x_i, x_i'') = \begin{cases} \frac{1-\eta(x_i'')}{|\mathcal{E}(x_{i-1}'')|-1} & \text{if } \nu(x_{i-1}'', x_i) = 1; \\ \frac{1}{|\mathcal{E}(x_{i-1}'')|} & \text{otherwise.} \end{cases}$$

We now claim $\eta(x_i'') \geq \frac{1}{|\mathcal{E}(x_{i-1}'')|} \geq \frac{1-\eta(x_i'')}{|\mathcal{E}(x_{i-1}'')|-1}$.

(1) First, we prove $\eta(x_i'') \geq \frac{1}{|\mathcal{E}(x_{i-1}'')|}$. One has

$$\begin{aligned}\eta(x_i'') - \frac{1}{|\mathcal{E}(x_{i-1}'')|} &= \frac{1}{(|\mathcal{E}(x_{i-1}'')| - 1)\exp(-\frac{\epsilon}{l}) + 1} - \frac{1}{|\mathcal{E}(x_{i-1}'')|} \\ &= \frac{(|\mathcal{E}(x_{i-1}'')| - 1)(1 - \exp(-\frac{\epsilon}{l}))}{((|\mathcal{E}(x_{i-1}'')| - 1)\exp(-\frac{\epsilon}{l}) + 1)|\mathcal{E}(x_{i-1}'')|}.\end{aligned}$$

For all $x_{i-1}''$ $(1 \leq i \leq k)$ contained in $\omega_3 = x_0'' x_1'' \cdots x_k'' \in X_{obs}^{k+1}$, $|\mathcal{E}(x_{i-1}'')| \geq 1$ holds.

Because $l \in \mathbb{N}^+$ and $\epsilon \geq 0$, we have $\frac{\epsilon}{l} \geq 0$. Further, the exponential function $\exp(-\frac{\epsilon}{l})$ has the range $0 < \exp(-\frac{\epsilon}{l}) \leq 1$.

We have $(|\mathcal{E}(x_{i-1}'')| - 1)(1 - \exp(-\frac{\epsilon}{l})) \geq 0$ due to $|\mathcal{E}(x_{i-1}'')| - 1 \geq 0$ and $1 - \exp(-\frac{\epsilon}{l}) \geq 0$. Similarly, thanks to $0 < \exp(-\frac{\epsilon}{l}) \leq 1$, $((|\mathcal{E}(x_{i-1}'')| - 1)\exp(-\frac{\epsilon}{l}) + 1)|\mathcal{E}(x_{i-1}'')| > 0$ holds.

To summarize, it is clear that

$$\eta(x_i'') - \frac{1}{|\mathcal{E}(x_{i-1}'')|} = \frac{(|\mathcal{E}(x_{i-1}'')| - 1)(1 - \exp(-\frac{\epsilon}{l}))}{((|\mathcal{E}(x_{i-1}'')| - 1)\exp(-\frac{\epsilon}{l}) + 1)|\mathcal{E}(x_{i-1}'')|} \geq 0.$$

(2) Let us prove the second inequality $\frac{1}{|\mathcal{E}(x_{i-1}'')|} \geq \frac{1-\eta(x_i'')}{|\mathcal{E}(x_{i-1}'')|-1}$.

$$\begin{aligned}&\frac{1}{|\mathcal{E}(x_{i-1}'')|} - \frac{1 - \eta(x_i'')}{|\mathcal{E}(x_{i-1}'')| - 1} \\ &= \frac{|\mathcal{E}(x_{i-1}'')| - 1 - |\mathcal{E}(x_{i-1}'')| + |\mathcal{E}(x_{i-1}'')|\eta(x_i'')}{|\mathcal{E}(x_{i-1}'')| \cdot (|\mathcal{E}(x_{i-1}'')| - 1)} \\ &= \frac{|\mathcal{E}(x_{i-1}'')|\eta(x_i'') - 1}{|\mathcal{E}(x_{i-1}'')| \cdot (|\mathcal{E}(x_{i-1}'')| - 1)} = \frac{\frac{(|\mathcal{E}(x_{i-1}'')|-1)\cdot(1-\exp(-\frac{\epsilon}{l}))}{(|\mathcal{E}(x_{i-1}'')|-1)\cdot\exp(-\frac{\epsilon}{l})+1}}{|\mathcal{E}(x_{i-1}'')| \cdot (|\mathcal{E}(x_{i-1}'')| - 1)} \\ &= \frac{1 - \exp(-\frac{\epsilon}{l})}{((|\mathcal{E}(x_{i-1}'')| - 1) \cdot \exp(-\frac{\epsilon}{l}) + 1) \cdot |\mathcal{E}(x_{i-1}'')|}\end{aligned}$$

Because $((|\mathcal{E}(x''_{i-1})|-1)\cdot\exp(-\frac{\epsilon}{l})+1)\cdot|\mathcal{E}(x''_{i-1})|>0$ and $1-\exp(-\frac{\epsilon}{l})\geq 0$, it follows that

$$\frac{1}{|\mathcal{E}(x''_{i-1})|}-\frac{1-\eta(x''_i)}{|\mathcal{E}(x''_{i-1})|-1}=\frac{1-\exp(-\frac{\epsilon}{l})}{[(|\mathcal{E}(x''_{i-1})|-1)\cdot\exp(-\frac{\epsilon}{l})+1]\cdot|\mathcal{E}(x''_{i-1})|}\geq 0.$$

Then, we conclude that

$$\begin{aligned}\frac{\mathbb{P}[\mathcal{M}_\omega(\omega_1)=\omega_3]}{\mathbb{P}[\mathcal{M}_\omega(\omega_2)=\omega_3]}&=\prod_{r\in R}\frac{\varphi(x''_{r-1},x_r,x''_r)}{\varphi(x''_{r-1},x'_r,x''_r)}\leq\prod_{r\in R}\frac{\eta(x''_r)}{\frac{1-\eta(x''_r)}{|\mathcal{E}(x''_{r-1})|-1}}\\&=\prod_{r\in R}\exp(\frac{\epsilon}{l})=\prod_{i=1}^{|R|}\exp(\frac{\epsilon}{l})\leq\exp(\epsilon).\end{aligned}$$

The constraint $\prod_{r\in R}\frac{\varphi(x''_{r-1},x_r,x''_r)}{\varphi(x''_{r-1},x'_r,x''_r)}\geq\exp(-\epsilon)$ can be proved similarly, which completes the proof. □

By Definition 4 and Theorem 1, Algorithm 2 computes an optimal mechanism for an unsafe input state sequence $\omega_1$, which aims to prevent the leakage of critical observability violative states or critical states in a non-critically observable system.

**Algorithm 2:** Construction of a mechanism $\mathcal{M}_\omega$.

**Input:** A non-critically observable NFA system $\mathcal{A}_{nd}=(X_{nd},E,\delta_{nd},X_0)$, a critical state estimator $Obs(\mathcal{A}_{nd})=(X_{obs},E_o,\delta_{obs},x_{0,obs})$ of $\mathcal{A}_{nd}$, a set of critical states $X_s\subseteq X_{nd}$, an unsafe input state sequence $\omega_1=x_0x_1\ldots x_k\in X_{obs}^{k+1}$, a real number $\epsilon\geq 0$, and an indicator function $\nu$.

**Output:** A mechanism $\mathcal{M}_\omega$.

$B:=G:=\emptyset$ $(B,G\subseteq X_{obs}^{k+1}\times\mathbb{N}\times\mathbb{N})$;
**for all** *productions* $\xi=x_0\xrightarrow{e_1}x_1\xrightarrow{e_2}\cdots\xrightarrow{e_k}x_k$ *in* $Obs(\mathcal{A}_{nd})$ *with* $F(\xi)=\omega_2$ **do**
  $b[1]:=\omega_2$;
  $b[2]:=deg(\omega_2)$;
  $b[3]:=\mathcal{J}(\omega_1,\omega_2)$;
  $B:=B\cup\{b\}$;
Compute the maximum number in set $B[2]$, i.e., $s=max\{b_1[2],\ldots,b_m[2]\}$ with $|B[2]|=m$;
**for all** $b\in B$ **do**
  **if** $b[2]=s$ **then**
    $G:=G\cup\{b\}$;
Compute the minimum number in set $G[3]$, i.e., $h=min\{g_1[3],\ldots,g_n[3]\}$ with $|G[3]|=n$;
**for all** $\omega=x_0x_1\cdots x_k\in X_{obs}^{k+1}$ **do**
  **for all** $\omega_3=x''_0x''_1\cdots x''_k\in X_{obs}^{k+1}$ **do**
    **for all** $i=1$ to $k$ **do**
      $\eta(x''_i):=\frac{1}{(|\mathcal{E}(x''_{i-1})|-1)\exp(-\frac{\epsilon}{h})+1}$;
    Invoke Algorithm 1 by inputting $Obs(\mathcal{A}_{nd})$, $\omega$, $\eta(x''_i)$, and $\nu$, which outputs the probability allocation strategy $\varphi$;
    Synthesize $\mathcal{M}_\omega$ by defining $\mathcal{M}_\omega(\omega)=\omega_3$ with $\mathbb{P}[\mathcal{M}_\omega(\omega)=\omega_3]=\prod_{i=1}^{k}\varphi(x''_{i-1},x_i,x''_i)$ $(\mathbb{P}[x''_0]=\mathbb{P}[x_0]=1)$;

Algorithm 2 first initializes two empty sets $B$ and $G$ for storing the triplet elements $b \in B$ and $g \in G$. Then, it iterates through all the productions $\xi = x_0 \xrightarrow{e_1} x_1 \xrightarrow{e_2} \cdots \xrightarrow{e_k} x_k$ with $F(\xi) = \omega_2$. In lines 3–6, for each such sequence $\omega_2$, the algorithm executes three key computations: assigning $\omega_2$, the degree $deg(\omega_2)$ of $\omega_2$, and the Hamming distance $\mathcal{J}(\omega_1,\omega_2)$ to $b[1]$ (We use a number in square brackets after a tuple to represent the entries of the tuple), $b[2]$, and $b[3]$, respectively. In line 7, the maximum value in set $B[2]$ (A number in square brackets after a set of tuples represents the set contained by the entries of the tuple set. Here, $B[2] = \{b_1[2], \ldots, b_m[2]\}$ with $b_i \in B (i = 1, \ldots, m)$) is computed and is stored in the variable $s$. Next, for each triplet $b$ in the set $B$, check whether $b[2]$ is equal to the maximum number $s$. If the condition is true, then add this element $b$ to the set $G$. In line 11, the value of the parameter $h$ is obtained by calculating the minimum number in set $G[3]$. For all input state sequences $\omega = x_0 x_1 \cdots x_k \in X_{obs}^{k+1}$ and for all $\omega_3 = x_0'' x_1'' \cdots x_k'' \in X_{obs}^{k+1}$, in line 15, we compute the correct probability $\eta(x_i'')$ for each output state $x_i''$. Finally, in lines 16–17, Algorithm 2 synthesizes the mechanism $\mathcal{M}_\omega$ by invoking the probability allocation strategy $\varphi$ of Algorithm 1.

The computational complexity of Algorithm 2 is mainly decided by the synthesis of the mechanism $\mathcal{M}_\omega$ in lines 12–17. In lines 12–13, we enumerate all input and output sequences, resulting in complexity $\mathcal{O}(|X_{obs}|^{2k})$. Furthermore, in lines 14–17, the computational complexity primarily depends on Algorithm 1, which has a value of $\mathcal{O}(k \times |X_{obs}|)$. Taking into account the above statements, the complexity is expressed as $\mathcal{O}(k \times |X_{obs}|^{3k})$.

**Theorem 2.** *Consider a critical state estimator $Obs(\mathcal{A}_{nd}) = (X_{obs}, E_o, \delta_{obs}, x_{0,obs})$ of a system $\mathcal{A}_{nd}$ that is not critically observable. Let $\omega_1 = x_0 x_1 \cdots x_k \in X_{obs}^{k+1}$ be an unsafe production state sequence in $Obs(\mathcal{A}_{nd})$. The mechanism $\mathcal{M}_\omega$ computed by Algorithm 2 guarantees DP of state sequences with respect to $\mathcal{W}_{k+1,h}$, in which $h \in \mathbb{N}$ is a parameter calculated by Algorithm 2. Moreover, the mechanism is optimal with respect to an unsafe production state sequence $\omega_1$.*

**Proof.** First, we prove that the mechanism $\mathcal{M}_\omega$ designed in Algorithm 2 provides DP of state sequences with respect to $\mathcal{W}_{k+1,h}$. Specifically, in lines 12–17, for all input state sequences $\omega$ and any possible output state sequence $\omega_3$, we have $\mathbb{P}[\mathcal{M}_\omega(\omega) = \omega_3] = \prod_{i=1}^{k} \varphi(x_{i-1}'', x_i, x_i'')$. Moreover, we accurately compute the correct state probability $\eta(x_i'')$ based on Equation (2). According to Theorem 1, the mechanism $\mathcal{M}_\omega$ provides DP of state sequences with respect to $\mathcal{W}_{k+1,h}$.

Then, we show that the mechanism $\mathcal{M}_\omega$ is optimal with respect to the unsafe state sequence $\omega_1$. In lines 2–6, for all productions $\xi$, the security degrees of their corresponding state sequences $F(\xi) = \omega_2$ are calculated, and these values are stored in the set $B[2]$. Then, the algorithm calculates the Hamming distance between $\omega_1$ and $\omega_2$, which reflects the degree of difference between them. In lines 7–10 of the algorithm, elements $b$ are assigned to $G$ such that $b[2] = s = deg(\omega_2) = deg_{max}(Obs(\mathcal{A}_{nd}), k+1)$. In line 11, for all $g[3] \in G[3]$, the minimum value $h$ in the set $G[3]$ is obtained, i.e., the minimum value $h$ of the Hamming distance between $\omega_1$ and $\omega_2$ is computed. To this end, there exists a state sequence $\omega_2$ with $deg(\omega_2) = deg_{max}(Obs(\mathcal{A}_{nd}), k+1)$ such that $(\omega_1, \omega_2) \in \mathcal{W}_{k+1,h}$, and there does not exist $h' < h$ such that $\mathcal{J}(\omega_1, \omega_2) = h'$ for all $\omega_2$ with the maximum degree. According to Definition 5, the mechanism $\mathcal{M}_\omega$ is optimal. □

In Algorithm 2, we design an optimal mechanism $\mathcal{M}_\omega$ with respect to an unsafe production state sequence $\omega_1$. When the mechanism $\mathcal{M}_\omega$ receives $\omega_1$ as input, a protected state sequence $\omega_3$ is probabilistically output. In addition, the mechanism guarantees that there exists at least one state sequence $\omega_2$ with the maximum security degree such that when $\omega_2$ is fed, it outputs $\omega_3$ with an approximate probability. Then, when an intruder obtains an observation $\rho' \in \mathcal{T}(\omega_3)$ derived from $\omega_3$, they are unable to infer whether $\omega_1$ or $\omega_2$ is generated in the critical state estimator $Obs(\mathcal{A}_{nd})$. Consequently, the disclosure of critical states or critical observability violative states becomes unattainable for the intruder. Furthermore, in order to maximize the obfuscation effect between $\omega_1$ and $\omega_2$, we especially

emphasize the optimization of the parameter $h$ in Algorithm 2, i.e., the Hamming distance between $\omega_1$ and $\omega_2$ is minimal.

**Example 5.** *We consider the NFA $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$ shown in Figure 3. Based on the results in Example 2 by assuming that the observable event set is $E_o = \{\alpha, \beta, \lambda\}$ and the secret is $X_s = \{x_3\}$, the system $\mathcal{A}_{nd}$ is not critically observable. Suppose that an unsafe observation $\rho = \beta\alpha\lambda$ is generated, in which the state estimate $\mathcal{C}(\rho) = x_{o4}$ is neither a subset of the critical states nor a subset of the non-critical states. The unsafe state sequence $\omega_1 = x_{0,obs}x_{o2}x_{o2}x_{o4} \in \mathcal{F}(\rho)$ corresponding to $\rho$ is obtained in the critical state estimator shown in Figure 4. By lines 2–6 of Algorithm 2, we calculate the set $B$ illustrated in Table 2 (the contents of columns 1, 2, and 3 record $B[1]$, $B[2]$, and $B[3]$, respectively). Then, we compute the maximum degree of the production state sequences in $B[1]$, i.e., $s = max\{b_1[2], b_2[2], \ldots, b_8[2]\} = 2$, and choose the minimum parameter $h = 3$.*

**Table 2.** Components of the set $B$ in Example 5.

| States Sequence | Security Degree | Hamming Distance |
|---|---|---|
| $x_{0,obs}x_{o2}x_{o2}x_{o2}$ | 0 | 1 |
| $x_{0,obs}x_{o2}x_{o4}x_{o3}$ | 1 | 2 |
| $x_{0,obs}x_{o2}x_{o2}x_{o3}$ | 1 | 1 |
| $x_{0,obs}x_{o2}x_{o3}x_{o5}$ | 1 | 2 |
| $x_{0,obs}x_{o2}x_{o4}x_{o4}$ | 0 | 1 |
| $x_{0,obs}x_{o2}x_{o4}x_{o2}$ | 0 | 2 |
| $x_{0,obs}x_{o1}x_{o3}x_{o1}$ | 2 | 3 |
| $x_{0,obs}x_{o1}x_{o3}x_{o5}$ | 1 | 3 |

*We take the input sequence $\omega_1 = x_{0,obs}x_{o2}x_{o2}x_{o4}$ and a random output $\omega_3 = x_{0,obs}x_{o1}x_{o3}x_{o5}$ as an example. The probabilities of the correct states are*

$$\eta(x_i'') = \frac{1}{(|\mathcal{E}(x_{i-1}'')| - 1)\exp(-\frac{\epsilon}{3}) + 1} \tag{7}$$

*with $i = \{1, 2, 3\}$, where in general $\epsilon \in [0.01, 10]$. Here, we consider $\epsilon = 0.1$.*

*By Equation (3), we have*

$$\begin{aligned}
&\eta(x_0'') = \eta(x_{0,obs}) = 1,\\
&\eta(x_1'') = \eta(x_{o2}) \approx 0.5083,\\
&\eta(x_2'') = \eta(x_{o2}) \approx 0.3408, \textit{and}\\
&\eta(x_3'') = \eta(x_{o4}) \approx 0.3408.
\end{aligned}$$

*The probability for generating $\omega_3$ is*

$$\begin{aligned}
&\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3]\\
&= \mathbb{P}[x_{0,obs}] \times \mathbb{P}[x_{o1}] \times \mathbb{P}[x_{o3}] \times \mathbb{P}[x_{o5}]\\
&= \eta(x_0'') \times \frac{1 - \eta(x_1'')}{2 - 1} \times \frac{1 - 0 \times \eta(x_2'')}{1 - 0} \times \frac{1 - 0 \times \eta(x_3'')}{2 - 0}\\
&\approx 1 \times (1 - 0.5083) \times 1 \times \frac{1}{2}\\
&\approx 0.2458.
\end{aligned}$$

*Analogously, for another random input state sequence (e.g.,* $\omega_2 = x_{0,obs}x_{o1}x_{o3}x_{o1}$ *with the maximum security degree), we have probabilities for the correct states*

$$\begin{aligned}
&\eta(x_0'') = \eta(x_{0,obs}) = 1,\\
&\eta(x_1'') = \eta(x_{o1}) \approx 0.5083,\\
&\eta(x_2'') = \eta(x_{o3}) = 1, \textit{and}\\
&\eta(x_3'') = \eta(x_{o1}) \approx 0.5083.
\end{aligned}$$

*When considering* $\omega_2$ *as an input of* $\mathcal{M}_\omega$*, the probability of generating* $\omega_3$ *is*

$$\begin{aligned}
&\mathbb{P}[\mathcal{M}_\omega(\omega_2) = \omega_3]\\
&= \mathbb{P}[x_{0,obs}] \times \mathbb{P}[x_{o1}] \times \mathbb{P}[x_{o3}] \times \mathbb{P}[x_{o5}]\\
&= \eta(x_0'') \times \eta(x_1'') \times \eta(x_2'') \times \frac{1-\eta(x_3'')}{2-1}\\
&\approx 1 \times 0.5038 \times 1 \times (1-0.5038)\\
&\approx 0.2499,
\end{aligned}$$

*and one has*

$$\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3] \leq \exp(\epsilon)\mathbb{P}[\mathcal{M}_\omega(\omega_2) = \omega_3].$$

*In other words, if the observation* $\rho = \beta\alpha\lambda$ *is directly sent, the critical observability violative state* $x_{o4}$ *can be detected by an intruder due to* $\mathcal{C}(\rho) = \{x_3, x_4, x_5\} \nsubseteq X_s$ *and* $\mathcal{C}(\rho) \nsubseteq X_{nd} \backslash X_s$*. However, by inputting* $\omega_1 = x_{0,obs}x_{o2}x_{o2}x_{o4} \in \mathcal{F}(\rho)$ *into the designed optimal DPM, a modified state sequence* $\omega_3 = x_{0,obs}x_{o1}x_{o3}x_{o5}$ *is output. Just by acquiring the sequence* $\omega_3$*, it is difficult for an intruder to backtrack or infer the original sequence* $\omega_1$*, as there exists another random sequence* $\omega_2 = x_{0,obs}x_{o1}x_{o3}x_{o1}$*, which can also generate* $\omega_3$ *with an approximate probability.*

## 5. Case Study

In this section, we consider a realistic case of enforcing critical observability in air traffic management (ATM) systems. In recent years, with the development of ATM systems, their network architecture has gradually shifted from closed, physically isolated networks to more interconnected and open cyber-physical systems. This evolution has made data exchange and information sharing efficient and convenient, but it has also created more possibilities for potential network attacks. Frequent and diverse types of network attacks pose a serious risk to the safety and stability of aviation. In this context, to strengthen security defenses by enhancing the system's awareness and response capability to potential threats, we propose an innovative solution: unsafe observations protection based on DP.

An ATM system is illustrated in Figure 6, detailing the communication process between the airport control tower and the pilot during the various stages of flight. On the one hand, the control tower sends a series of instructions to the pilot, such as pre-flight/stop-flight instructions, engine start instructions, etc. These instructions guide the pilot's actions and determine that the aircraft is airworthy and all parts of the system are operating normally. On the other hand, the pilot's role is to wait for and follow instructions from the control tower, performing the corresponding operations. If there are any faults or abnormal situations during the flight, the pilot must immediately report to the tower. This real-time feedback mechanism allows the control tower to take action quickly to resolve issues that may threaten flight safety. The exchange of information between the control tower and the pilot runs throughout the entire flight process and jointly affects the aircraft's maneuvers (represented by the downward arrows in Figure 6). A complete aerial flight can be divided into five stages: taxiing, climbing, cruising, descending, and landing. Each stage requires precise coordination between the pilot and the control tower. Based on Figure 6, we model an ATM system as an NFA (shown in Figure 7) by using a single aircraft as a study case and define a series of important physical states as critical states. The meanings of states

and events in the NFA are listed in Table 3. We further assume that an intruder launches an attack as soon as they capture the critical states or critical observability violative states. To prevent the system $\mathcal{A}_{nd}$ from being corrupted, we introduce a DPM, such that the unsafe states are not accurately captured by an intruder. From a practical standpoint, this research establishes the groundwork for subsequent security defense of cyber-physical systems and has significant implications for safeguarding air transportation safety and maintaining airspace order.

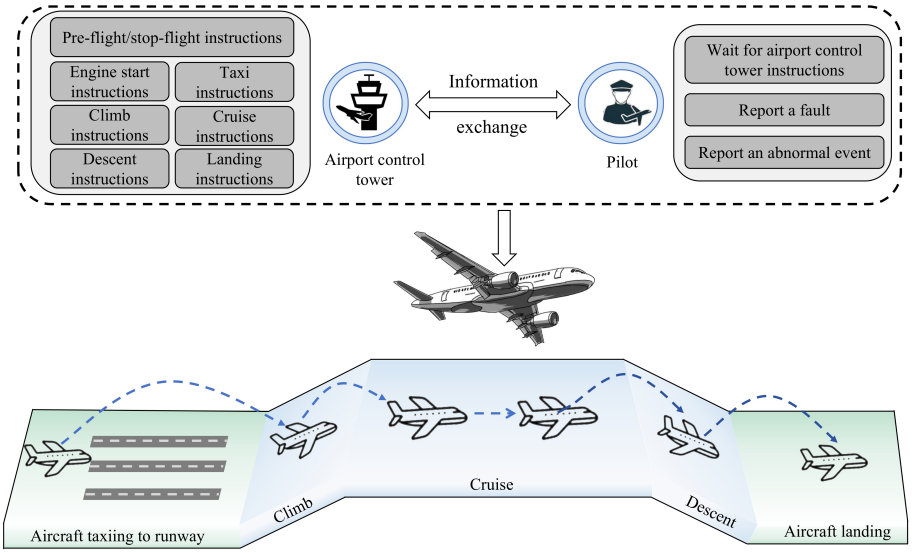



**Figure 6.** Air traffic management systems.

**Table 3.** Meanings of states and events of the NFA portrayed in Figure 7.

| State | Meanings of States | Event | Meanings of Events |
|---|---|---|---|
| $x_0$ | Pilot boards an aircraft and waits for instructions | $a$ | Send a pre-flight instruction |
| $x_1$ | Pilot executes a pre-flight check | $b$ | Incomplete pre-flight check |
| $x_2$ | Waiting for the engine to start | $c$ | Complete pre-flight check |
| $x_3$ | Fault reports are received | $d$ | Send a fault report |
| $x_4$ | Waiting for taxi clearance | $e$ | Reject engine start |
| $x_5$ | Rejected take off | $f$ | Allow engine start |
| $x_6$ | Aircraft taxiing to runway $A_1$ | $g$ | Send stop-flight instructions |
| $x_7$ | Aircraft taxiing to runway $A_2$ | $h$ | Obtain taxi clearance |
| $x_8$ | Aircraft taxiing to runway $A_3$ | $i$ | Send a climb request |
| $x_9$ | Aircraft waiting for authorization at the runway | $j$ | Climb request granted |
| $x_{10}$ | Aircraft take off | $k$ | Send a cruise instruction |
| $x_{11}$ | Aircraft cruise | $l$ | Climb request rejected |
| $x_{12}$ | Stop flight and return to origin | $m$ | Send an anomaly report |
| $x_{13}$ | Aircraft started without authorization | $n$ | Taxi clearance not granted |
| $x_{14}$ | Aircraft descent | $o$ | Send a descent instruction |
| $x_{15}$ | Aircraft landing | $p$ | Send a landing instruction |

In Figure 7, the pilot initially executes boarding and waits for instructions from the airport control tower (state $x_0$). When the pilot receives a pre-flight instruction sent by the tower (event $a$), the state is transferred from $x_0$ to $x_1$, at which point the pilot executes a pre-flight check. For any error found by the pilot during the check, a fault is reported (event $d$) to the tower, and the state $x_3$ is reached. In state $x_2$, the aircraft is waiting for the engine to start. If the engine fails to start (event $e$) due to safety issues or other conditions, the state remains in $x_2$; otherwise, the state transitions to $x_4$, where the pilot communicates with the tower in the hope of obtaining a taxi clearance. The taxiing process is represented by states $x_6$, $x_7$, and $x_8$, which correspond to different runways. The event $h$, obtaining

taxi clearance, triggers the transition from $x_4$ to one of these taxing states. Once the taxi clearance is not granted (event $n$), the state transitions from $x_4$ to $x_{13}$ or $x_5$.

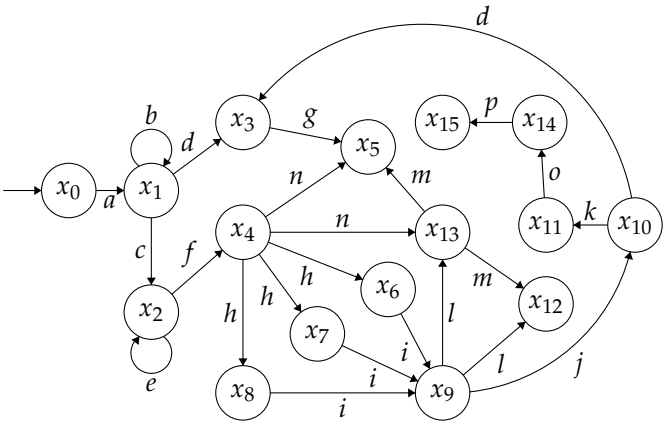



**Figure 7.** An NFA model $\mathcal{A}_{nd}$ of an air traffic management system.

The event $i$, sending a climbing request, triggers the transition from one of the taxiing states ($x_6$, $x_7$, or $x_8$) to state $x_9$. If the climb request is granted (event $j$), the aircraft takes off and leaves the ground (state $x_{10}$). In addition, fault reports (event $d$) can lead to a transition from $x_{10}$ to $x_3$, allowing the aircraft to report any troubles encountered during the flight. However, if the climb request is rejected (event $l$), the state is switched from $x_9$ to $x_{12}$, in which case the flight is stopped and the aircraft returns to its origin. In exceptional cases, the aircraft may be started without authorization, resulting in a move from $x_9$ to $x_{13}$, as triggered by event $l$. In $x_{13}$, the pilot sends an anomaly report to the tower (event $m$), where the aircraft refuses to execute the flight mission and $x_{12}$ or $x_5$ is reached.

**Example 6.** *Let us consider the NFA* $\mathcal{A}_{nd} = (X_{nd}, E, \delta_{nd}, X_0)$ *portrayed in Figure 7, where* $E = E_o \dot{\cup} E_{uo}$ *is the set of events with* $E_o = \{a, b, c, d, e, f, g, h, i, j, k, l, n, o, p\}$ *and* $E_{uo} = \{m\}$, *and* $X_0 = \{x_0\}$ *is the set of initial states. We assume that the set of critical states is* $X_s = \{x_5, x_{12}\}$. *Using the standard determinization procedure presented in [28], we construct a critical state estimator* $Obs(\mathcal{A}_{nd}) = \{X_{obs}, E_o, \delta_{obs}, x_{0,obs}\}$ *of* $\mathcal{A}_{nd}$, *which is shown in Figure 8. The components of all states are listed in Table 4.*

*Given an observation* $\rho = acfn$, *we conclude that the critical observability is violated thanks to* $\mathcal{C}(\rho) = \delta_{obs}(x_{0,obs}, acfn) = x_{o7} = \{x_5, x_{12}, x_{13}\} \nsubseteq X_s$ *and* $\mathcal{C}(\rho) \nsubseteq X_{nd} \backslash X_s$. *Therefore,* $\rho$ *is an unsafe observation and* $\omega_1 = x_{0,obs}x_{o1}x_{o2}x_{o4}x_{o7} \in \mathcal{F}(\rho)$ *is an unsafe production state sequence.*

*We now show how the unsafe production state sequence is protected by designed optimal mechanisms* $\mathcal{M}_\omega$ *that guarantees DP. Consider* $\omega_1$ *as an input of* $\mathcal{M}_\omega$. *By Algorithm 2, for all* $\omega_2 \in X^5_{obs}$, *the security degrees* $deg(\omega_2)$ *and Hamming distances* $\mathcal{J}(\omega_1, \omega_2)$ *are listed in Table 5, where the first column represents all production state sequences of length 5 in* $Obs(\mathcal{A}_{nd})$, *the second column indicates the security degree of the sequence, and the third column shows the Hamming distance between the sequence and* $\omega_1$. *If* $\omega_2 = x'_0x'_1 \cdots x'_k$ *is a safe production state sequence, the security degree of* $\omega_2$ *is* $deg(\omega_2) = |x'_k|$. *Otherwise, if* $\omega_2$ *is unsafe, we have* $deg(\omega_2) = 0$. *The Hamming distance between* $\omega_1$ *and* $\omega_2$ *is calculated according to Definition 3. For example, in Table 5, for a safe production state sequence* $\omega_2 = x_{0,obs}x_{o1}x_{o2}x_{o4}x_{o6}$, *we have* $deg(\omega_2) = |x_{o6}| = |\{x_6, x_7, x_8\}| = 3$. *The Hamming distance between* $\omega_1$ *and* $\omega_2$ *is* $\mathcal{J}(\omega_1, \omega_2) = 1$.

*Then, based on Table 5, we compute the maximum degree* $s = 3$ *and for all* $\omega_2$ *with* $deg(\omega_2) = deg_{max}(Obs(\mathcal{A}_{nd}), 5) = 3$, *we select the minimum parameter* $h = 1$. *For all* $(\omega_1, \omega_2) \in \mathcal{W}_{5,1}$, *and for all* $\omega_3 \in X^5_{obs}$, *it holds that*

$$\mathbb{P}[\mathcal{M}_\omega(\omega_1) = \omega_3] \leq \exp(\epsilon)\mathbb{P}[\mathcal{M}_\omega(\omega_2) = \omega_3],$$

*where in general* $\epsilon \in [0.01, 10]$. *Here, we consider* $\epsilon = 0.1$.

*When observing $\rho = acfn$, the state that violates critical observability is detectable by an intruder. However, by applying the optimal mechanism designed by Algorithm 2 that outputs a randomly modified production state sequence $\omega_3$, the unsafe sequence $\omega_1 \in \mathcal{F}(\rho)$ cannot be ensured, as another sequence $\omega_2$ with $(\omega_1, \omega_2) \in \mathcal{W}_{5,1}$ can generate $\omega_3$ with a similar probability. In this way, unsafe observations in the ATM system cannot be accurately captured. Thereby, the intruder is unable to launch an attack to the ATM system $\mathcal{A}_{nd}$. The computational complexity of the optimal mechanism $\mathcal{M}_\omega$ is $\mathcal{O}(k \times |X_{obs}|^{3k}|)$ where, in the ATM system, $k = 5$ and $X_{obs} = 13$.*

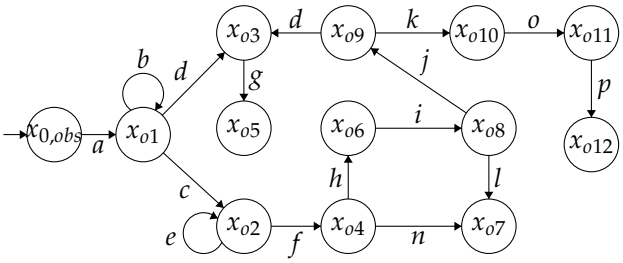



**Figure 8.** The critical state estimator $Obs(\mathcal{A}_{nd})$ of the air traffic management system $\mathcal{A}_{nd}$.

**Table 4.** Components of the states in the estimator shown in Figure 8.

| State | State Estimation | State | State Estimation |
|---|---|---|---|
| $x_{0,obs}$ | $\{x_0\}$ | $x_{o7}$ | $\{x_5, x_{12}, x_{13}\}$ |
| $x_{o1}$ | $\{x_1\}$ | $x_{o8}$ | $\{x_9\}$ |
| $x_{o2}$ | $\{x_2\}$ | $x_{o9}$ | $\{x_{10}\}$ |
| $x_{o3}$ | $\{x_3\}$ | $x_{o10}$ | $\{x_{11}\}$ |
| $x_{o4}$ | $\{x_4\}$ | $x_{o11}$ | $\{x_{14}\}$ |
| $x_{o5}$ | $\{x_5\}$ | $x_{o12}$ | $\{x_{15}\}$ |
| $x_{o6}$ | $\{x_6, x_7, x_8\}$ | | |

**Table 5.** All production state sequences with length 5 in $Obs(\mathcal{A}_{nd})$.

| States Sequence | Security Degree | Hamming Distance |
|---|---|---|
| $x_{0,obs}x_{o1}x_{o1}x_{o1}x_{o1}$ | 1 | 3 |
| $x_{0,obs}x_{o1}x_{o1}x_{o1}x_{o2}$ | 1 | 3 |
| $x_{0,obs}x_{o1}x_{o1}x_{o1}x_{o3}$ | 1 | 3 |
| $x_{0,obs}x_{o1}x_{o1}x_{o3}x_{o5}$ | 0 | 3 |
| $x_{0,obs}x_{o1}x_{o1}x_{o2}x_{o2}$ | 1 | 3 |
| $x_{0,obs}x_{o1}x_{o1}x_{o2}x_{o4}$ | 1 | 3 |
| $x_{0,obs}x_{o1}x_{o2}x_{o2}x_{o2}$ | 1 | 2 |
| $x_{0,obs}x_{o1}x_{o2}x_{o2}x_{o4}$ | 1 | 2 |
| $x_{0,obs}x_{o1}x_{o2}x_{o4}x_{o6}$ | 3 | 1 |

## 6. Conclusions

This work proposes a differential privacy (DP) mechanism for the enforcement of critical observability in DESs modeled with finite-state automata. Meanwhile, the mechanism protects the predefined critical states from being captured by an intruder. We first recall the formalization of critical observability verification problems. Specifically, given an NFA system $\mathcal{A}_{nd}$, a set of observable events $E_o$ and a predefined set of critical states $X_s$, to verify whether the system $\mathcal{A}_{nd}$ is critically observable with respect to $E_o$ and $X_s$, one can construct a critical state estimator $Obs(\mathcal{A}_{nd})$ of $\mathcal{A}_{nd}$. In $Obs(\mathcal{A}_{nd})$, the system is considered non-critically observable if there exists a state that is neither a subset of the critical states nor a subset of the noncritical states. Given a non-critically observable system, we define unsafe observations that violate critical observability or reveal the critical states and their corresponding unsafe production state sequences. To guarantee that unsafe production state sequences are not detected by an intruder, we introduce a mechanism called state sequence DP. Furthermore, in order to most effectively protect unsafe production state

sequences, we define an optimal mechanism and design an algorithm for synthesizing the optimal mechanism. Finally, we take the ATM system as an example to delve into the implementation of critical state protection in this system. We provide a detailed explanation of the modeling process of the ATM system and define certain important physical states as critical states. Additionally, we demonstrate how to effectively protect unsafe observations using the differential privacy mechanism.

From a practical perspective, this research has potential applications in various cyber-physical systems. In the future, we will consider applying the differential privacy mechanism to the verification and enforcement of other properties such as $k$-step and infinite-step opacity. Moreover, it is worth exploring the critical observability problem when a system is subject to sensor attacks, i.e., sensor readings can altered by an external attacker.

**Author Contributions:** Conceptualization, J.Z. and Z.L.; Methodology, J.Z.; Writing—original draft, J.Z.; Writing—review & editing, Z.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is partially supported by the Science and Technology Development Fund (FDCT) under grant 2023/0029/RIA1.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

## References

1. Lin, F. Opacity of discrete event systems and its applications. *Automatica* **2011**, *47*, 496–503. [CrossRef]
2. Yang, J.; Deng, W.; Qiu, D.; Jiang, C. Opacity of networked discrete event systems. *Inf. Sci.* **2021**, *543*, 328–344. [CrossRef]
3. Deng, W.; Qiu, D.; Yang, J. Opacity measures of fuzzy discrete Event Systems. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 2612–2622. [CrossRef]
4. Cao, L.; Shu, S.; Lin, F.; Chen, Q.; Liu, C. Weak diagnosability of discrete-event systems. *IEEE Trans. Control. Netw. Syst.* **2022**, *9*, 184–196. [CrossRef]
5. Liu, F.; Qiu, D. Safe diagnosability of stochastic discrete event systems. *IEEE Trans. Autom. Control.* **2008**, *53*, 1291–1296. [CrossRef]
6. Balun, J.; Masopust, T. On verification of D-detectability for discrete event systems. *Automatica* **2021**, *133*, 109884. [CrossRef]
7. Zhu, H.; Liu, G.; Yu, Z.; Li, Z. Detectability in discrete event systems using unbounded Petri nets. *Mathematics* **2023**, *11*, 3862. [CrossRef]
8. Li, Y.; Wang, M.; Jones, A. An approach for the design of supervisory controller of discrete event systems. In Proceedings of the 3rd World Congress on Intelligent Control and Automation, Hefei, China, 28 June–2 July 2000; pp. 2341–2346.
9. De Santis, E.; Di Benedetto, M.D.; Di Gennaro, S.; D'Innocenzo, A.; Pola, G. Critical observability of a class of hybrid systems and application to air traffic management. In *Stochastic Hybrid Systems: Theory and Safety Critical Applications*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 141–170.
10. Zhu, G.; Yin, L.; Li, Y.; Li, Z.; Wu, N. Identification of labeled Petri nets from finite automata. *Inf. Sci.* **2024**, *667*, 120448. [CrossRef]
11. Di Benedetto, M.D.; Di Gennaro, S.; D'Innocenzo, A. Discrete state observability of hybrid systems. *Int. J. Robust Nonlinear Control.* **2009**, *19*, 1564–1580. [CrossRef]
12. Tong, Y.; Ma, Z. Verification of $k$-step and definite critical observability in discrete-event systems. *IEEE Trans. Autom. Control.* **2023**, *68*, 4305–4312. [CrossRef]
13. Pola, G.; De Santis, E.; Di Benedetto, M.D.; Pezzuti, D. Design of decentralized critical observers for networks of finite state machines: A formal method approach. *Automatica* **2017**, *86*, 174–182. [CrossRef]
14. Lai, A.; Lahaye, S.; Komenda, J. Observer construction for polynomially ambiguous max-plus automata. *IEEE Trans. Autom. Control.* **2022**, *67*, 1582–1588. [CrossRef]
15. Cong, X.; Fanti, M.P.; Mangini, A.M.; Li, Z. Critical observability of Petri nets with unknown initial marking. In Proceedings of the 58th IEEE Conference on Decision and Control (CDC), Nice, France, 11–13 December 2019; pp. 2003–2008.
16. Cong, X.; Fanti, M.P.; Mangini, A.M.; Li, Z. Critical observability of safe Petri nets via integer linear programming. In Proceedings of the 57th IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 1029–1034.
17. Dwork, C. Differential privacy. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Venice, Italy, 10–14 July 2006; pp. 1–12.
18. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In Proceedings of the 3rd Theory of Cryptography Conference, New York, NY, USA, 4–7 March 2006; pp. 265–284.

19. McSherry, F.; Talwar, K. Mechanism design via differential privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, Providence, RI, USA, 21–23 October 2007; pp. 94–103.
20. Lin, C.; Wang, P.; Song, H.; Zhou, Y.; Liu, Q.; Wu, G. A differential privacy protection scheme for sensitive big data in body sensor networks. *Ann. Telecommun.* **2016**, *71*, 465–475. [CrossRef]
21. Li, W.; Wang, S.; Wang, H.; Lu, Y. Child health dataset publishing and mining based on differential privacy preservation. *Mathematics* **2024**, *12*, 2487. [CrossRef]
22. Zhao, Y.; Zhao, J.; Yang, M.; Wang, T.; Wang, N.; Lyu, L.; Niyato, D.; Lam, K.Y. Local differential privacy-based federated learning for internet of things. *IEEE Internet Things J.* **2021**, *8*, 8836–8853. [CrossRef]
23. Ren, J.; Jiang, L.; Peng, H.; Lyu, L.; Liu, Z.; Chen, C.; Wu, J.; Bai, X.; Yu, P.S. Cross-network social user embedding with hybrid differential privacy guarantees. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 1685–1695.
24. Jones, A.; Leahy, K.; Hale, M. Towards differential privacy for symbolic systems. In Proceedings of the 2019 American Control Conference, Philadelphia, PA, USA, 10–12 July 2019; pp. 372–377.
25. Chen, B.; Leahy, K.; Jones, A.; Hale, M. Differential privacy for symbolic systems with application to Markov Chains. *Automatica* **2023**, *152*, 110908. [CrossRef]
26. Kang, H.; Zhang, S.; Jia, Q. A method for time-series location data publication based on differential privacy. *Wuhan Univ. J. Nat. Sci.* **2019**, *24*, 107–115. [CrossRef]
27. Hua, J.; Gao, Y.; Zhong, S. Differentially private publication of general time-serial trajectory data. In Proceedings of the 2015 IEEE Conference on Computer Communications, Kowloon, Hong Kong, 26 April–1 May 2015.
28. Cassandras, C.G.; Lafortune, S. *Introduction to Discrete Event Systems*; Springer: Berlin/Heidelberg, Germany, 2009.
29. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–406. [CrossRef]
30. Cong, X.; Fanti, M.P.; Mangini, A.M.; Li, Z. Critical observability verification and enforcement of labeled Petri nets by using basis markings. *IEEE Trans. Autom. Control.* **2023**, *68*, 8158–8164. [CrossRef]
31. Xie, Y.; Yin, X.; Li, S. Opacity enforcing supervisory control using nondeterministic supervisors. *IEEE Trans. Autom. Control.* **2021**, *67*, 6567–6582. [CrossRef]
32. Yang, S.; Hou, J.; Yin, X.; Li, S. Opacity of networked supervisory control systems over insecure communication channels. *IEEE Trans. Control. Netw. Syst.* **2021**, *8*, 884–896. [CrossRef]
33. Li, X.; Hadjicostis, C.N.; Li, Z. Extended insertion functions for opacity enforcement in discrete-event systems. *IEEE Trans. Autom. Control.* **2021**, *67*, 5289–5303. [CrossRef]
34. Mohajerani, S.; Ji, Y.; Lafortune, S. Compositional and abstraction-based approach for synthesis of edit functions for opacity enforcement. *IEEE Trans. Autom. Control.* **2019**, *65*, 3349–3364. [CrossRef]
35. Wu, Y.C.; Lafortune, S. Synthesis of insertion functions for enforcement of opacity security properties. *Automatica* **2014**, *50*, 1336–1348. [CrossRef]
36. Hamming, R.W. Error detecting and error correcting codes. *Bell Syst. Tech. J.* **1950**, *29*, 147–160. [CrossRef]
37. Zhang, J.; Dong, Y.; Yin, L.; Mostafa, A.M.; Li, Z. Opacity enforcement in discrete event systems using differential privacy. *Inf. Sci.* **2025**, *688*, 121284. [CrossRef]


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.