On Verification of Weak and Strong k-Step Opacity for Discrete-Event Systems ⋆

Jiří Balun and **Tomáš Masopust**

Faculty of Science, Palacky University, Olomouc, Czechia
(e-mails: jiri.balun01@upol.cz, tomas.masopust@upol.cz)

Abstract: Opacity is an important property asking whether a passive observer (an intruder), who knows the structure of the system but has only a limited observation of its behavior, may reveal the secret of the system. Several notions of opacity have been studied in the literature, including current-state opacity, k-step opacity, and infinite-step opacity. We investigate weak and strong k-step opacity, the notions that generalize both current-state opacity and infinite-step opacity, and ask whether the intruder is not able to decide, at any instant, when respectively whether the system was in a secret state during the last k observable steps. We design a new algorithm to verify weak k-step opacity, the complexity of which is lower than that of existing algorithms and that does not depend on the parameter k. Then, we show how to use this algorithm to verify strong k-step opacity by reducing the verification of strong k-step opacity to the verification of weak k-step opacity. The complexity of the resulting approach is again better than that of existing algorithms, and does not depend on the parameter k.

Keywords: Discrete event systems, finite automata, opacity, verification, complexity

1. INTRODUCTION

Opacity is an important information flow property that is used to study security and privacy questions of discrete-event systems, communication protocols, or computer systems. It guarantees that a system prevents an intruder from revealing its secret. The intruder is a passive observer that knows the structure of the system but that has only a limited observation of system's behavior.

The secret is modeled as a set of secret states or as a set of secret behaviors. Modeling the secret as a set of secret states results in state-based opacity, introduced by Bryans et al. (2005, 2008) for Petri nets and transition systems, and later adapted to automata by Saboori and Hadjicostis (2007). Modeling the secret as a set of secret behaviors results in language-based opacity, introduced by Badouel et al. (2007) and Dubreil et al. (2008). We refer the reader to the overview by Jacob et al. (2016) for more details.

Many different notions of opacity have been discussed in the literature, including initial-state opacity and current-state opacity. While initial-state opacity prevents the intruder from revealing, at any instant, whether the system started in a secret state, current-state opacity prevents the intruder "only" from revealing whether the current state of the system is secret. The intruder may, however, later realize that the system was in a secret state at a former step of the computation. For example, if the intruder estimates that the system is in one of two possible states and, in the next step, the system proceeds by an observable event enabled only from one of the states, then the intruder reveals the state in which the system was one step ago.

This problem motivated Saboori and Hadjicostis (2007, 2012) to introduce the notion of weak k-step opacity, which requires that the intruder is not able to ascertain the secret in the current state and k subsequent observable steps. Notice that the cases of $k = 0$ and $k = \infty$ coincide with the notions of current-state opacity and infinite-step opacity, respectively, discussed in the literature.

The verification of weak k-step opacity has been intensively studied in the literature. In particular, there are five main approaches to verify weak k-step opacity based on the secret observer with complexity $O(\ell 2^{n(k+3)})$, where n is the number of states and ℓ is the number of observable events, on the reverse comparison with complexity $O((n+m)(k+1)3^n)$, where $m \leq \ell n^2$, on the state estimator of Saboori and Hadjicostis (2011) with complexity $O(\ell(\ell+1)^k 2^n)$, on the two-way observer of Yin and Lafortune (2017) with complexity $O(\min\{n2^{2n}, n\ell^k 2^n\})$, including a minor correction by Lan et al. (2020), and on the projected automaton of Balun and Masopust (2021) with complexity $O((k+1)2^n(n+m\ell^2))$; see also Wintenberg et al. (2022) for more details on the state complexity and an experimental comparison.

The reader can see that the complexity of all the above algorithms depends on the parameter k. A partial exception is the two-way observer, which does not depend on the parameter k if $\ell^k > 2^n$, that is, if k is larger than the proportion of states to the logarithm of the number of observable events.

In this paper, we design a new algorithm to verify weak k-step opacity, the complexity bound of which does not depend on the parameter k. In particular, the state com-

⋆ Supported by MEYS under the INTER-EXCELLENCE project LTAUSA19098 and by Palacky University project IGA PrF 2022 018. T. Masopust is also with Institute of Mathematics of the Czech Acad. Sci. supported by GAČR grant GC19-06175J and by RVO 67985840.

plexity of our algorithm is $O(n2^n)$ and the time complexity is $O((n+m)2^n)$, where n is the number of states of the input automaton and $m \leq \ell n^2$ is the number of transitions of the projected input automaton. Our algorithm is, therefore, in general faster than the existing algorithms.

Indeed, the exception is for the special case of a very small parameter k; namely, if k is smaller than $2\log(n)/\log(\ell)$, where n is the number of states of the input automaton and ℓ is its number of observable events, then the algorithms based on the state estimator of Saboori and Hadjicostis (2011) and on the two-way observer of Yin and Lafortune (2017) are, in the worst-case, faster than our algorithm.

Later, Falcone and Marchand (2014) have shown that even weak k-step opacity may not be as confidential as intuitively expected. Namely, the intruder may realize that the system was in a secret state, though it cannot deduce the exact time when it happened (see an example in Section 4 or in Falcone and Marchand (2014)).

This problem motivated Falcone and Marchand (2014) to introduce the notion of strong k-step opacity as the notion of k-step opacity with a higher level of confidentiality. The idea is that, whereas weak k-step opacity prevents the intruder from revealing the exact time when the system was in a secret state during the last k observable steps, strong k-step opacity prevents the intruder from revealing that the system was in a secret state during the last k observable steps.

Notice that the literature so far (and so do we) consider only deterministic DES where states that are not secret are nonsecret. In this settings, strong k-step opacity implies weak k-step opacity. However, the verification of one type of k-step opacity cannot be directly used for the verification of the other. Nonetheless, we show how to do it indirectly by constructing a polynomial-time transformation of an instance of the strong k-step opacity problem to an instance of the weak k-step opacity problem, which allows us to verify strong k-step opacity by the algorithms for weak k-step opacity.

In particular, with the help of our new algorithm verifying weak k-step opacity described in Section 3, we obtain a new algorithm for the verification of strong k-step opacity with a lower complexity than that of existing algorithms that, in addition, does not depend on the parameter k.

An extended version containing all the missing details and proofs will be available at the preprint sever arXiv (Balun and Masopust, 2022).

2. PRELIMINARIES

We assume that the reader is familiar with discrete-event systems (Cassandras and Lafortune, 2021). For a set S, $|S|$ denotes the cardinality of S, and 2^S denotes the power set of S. An alphabet Σ is a finite nonempty set of events. A string over Σ is a sequence of events from Σ; the empty string is denoted by ε. The set of all finite strings over Σ is denoted by Σ^*. A language L over Σ is a subset of Σ^*. The set of all prefixes of strings of L is the set $\overline{L} = \{u \mid \text{there is } v \in \Sigma^* \text{ such that } uv \in L\}$. For a string $u \in \Sigma^*$, $|u|$ denotes the length of u.

A *nondeterministic finite automaton* (NFA) over an alphabet Σ is a structure $G = (Q, \Sigma, \delta, I, F)$, where Q is a finite set of states, $I \subseteq Q$ is a nonempty set of initial states, $F \subseteq Q$ is a set of marked states, and $\delta\colon Q \times \Sigma \to 2^Q$ is a transition function that can be extended to the domain $2^Q \times \Sigma^*$ by induction. In addition, for a language $S \subseteq \Sigma^*$, we define $\delta(Q, S) = \cup_{s \in S} \delta(Q, s)$. The language marked by G is the set $L_m(G) = \{w \in \Sigma^* \mid \delta(I, w) \cap F \neq \emptyset\}$, and the language generated by G is the set $L(G) = \{w \in \Sigma^* \mid \delta(I, w) \neq \emptyset\}$.

The NFA G is *deterministic* (DFA) if $|I| = 1$ and $|\delta(q, a)| \leq 1$ for every $q \in Q$ and $a \in \Sigma$. We also identify the singleton $I = \{q_0\}$ with its element q_0, and simply write $G = (Q, \Sigma, \delta, q_0, F)$ instead of $G = (Q, \Sigma, \delta, \{q_0\}, F)$.

A *discrete-event system* (DES) G over Σ is an NFA over Σ together with the partition of Σ into Σ_o and Σ_{uo} of *observable* and *unobservable events*, respectively. If we want to specify that the DES is modeled by a DFA, we talk about a *deterministic* DES. If the marked states are irrelevant, we omit them and simply write $G = (Q, \Sigma, \delta, I)$.

State estimation is modeled by *projection* $P\colon \Sigma^* \to \Sigma_o^*$, which is a morphism for concatenation defined by $P(a) = \varepsilon$ if $a \in \Sigma_{uo}$, and $P(a) = a$ if $a \in \Sigma_o$. The action of P on a string $a_1 a_2 \cdots a_n$ is to erase all unobservable events, that is, $P(a_1 a_2 \cdots a_n) = P(a_1)P(a_2)\cdots P(a_n)$. The definition can be readily extended to languages.

Let G be a DES over Σ, and let $P\colon \Sigma^* \to \Sigma_o^*$ be the corresponding projection. The *projected automaton* of G is the NFA $P(G)$ obtained from G by replacing every transition (p, a, q) by $(p, P(a), q)$, and by the standard elimination of ε-transitions. In particular, if δ is the transition function of G, then the transition function $\gamma\colon Q \times \Sigma_o \to 2^Q$ of $P(G)$ is defined as $\gamma(q, a) = \delta(q, P^{-1}(a))$. Then, the projected automaton $P(G)$ is an NFA over Σ_o with the same states as G that recognizes the language $P(L_m(G))$ and that can be constructed in polynomial time (Hopcroft et al., 2006).

We call the DFA constructed from $P(G)$ by the standard subset construction a *full observer* of G. The accessible part of the full observer of G is called an *observer* of G in the literature, cf. (Cassandras and Lafortune, 2021). The full observer has exponentially many states compared with G. In the worst case, the same holds for the observer as well, cf. Jirásková and Masopust (2012); Wong (1998).

Transitions of the product of two DESs do not depend on the initial states of the particular automata (Hopcroft et al., 2006). Therefore, we disregard the initial states in the considered DESs and simply write two DESs over the alphabet Σ as $G_i = (Q_i, \Sigma, \delta_i)$, for $i = 1, 2$. The *product automaton* of G_1 and G_2 is defined as the DES $G_1 \times G_2 = (Q_1 \times Q_2, \Sigma, \delta)$, where $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$, for every $(q_1, q_2) \in Q_1 \times Q_2$ and $a \in \Sigma$. This definition captures the concept of the product automaton, and allows us to take any subset of $Q_1 \times Q_2$ as initial states of $G_1 \times G_2$, which is useful in our algorithm.

3. VERIFICATION OF WEAK K-STEP OPACITY

In this section, we design a new algorithm for the verification of weak k-step opacity. Compared with the existing algorithms, our algorithm does not depend on the param-

eter k, and its complexity is, in general, lower than that of existing algorithms.

Before we recall the definition of weak k-step opacity, we denote the set of all nonnegative integers with their limit by $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$. Then, for $k \in \mathbb{N}_\infty$, weak k-step opacity is a property whether the intruder is not able to reveal the secret of a system in the current and k subsequent states.

Definition 1. Given a DES $G = (Q, \Sigma, \delta, I)$ and $k \in \mathbb{N}_\infty$. System G is *weakly k-step opaque (k-SO)* with respect to the sets Q_S of secret and Q_{NS} of nonsecret states and projection $P\colon \Sigma^* \to \Sigma_o^*$ if for every string $st \in L(G)$ with $|P(t)| \leq k$ and $\delta(\delta(I, s) \cap Q_S, t) \neq \emptyset$, there exists a string $s't' \in L(G)$ such that $P(s) = P(s')$, $P(t) = P(t')$, and $\delta(\delta(I, s') \cap Q_{NS}, t') \neq \emptyset$.

The complexity of existing algorithms verifying weak k-step opacity is exponential and depends on the parameter k. The exponential complexity seems unavoidable because the problem is PSPACE-complete, see Balun and Masopust (2021) for more details, and it is a long-standing open problem of computer science if PSPACE-complete problems can be solved in polynomial time.

We now design our algorithm, described as Algorithm 1, verifying weak k-step opacity with the time complexity $O((n + m)2^n)$, where n is the number of states of the automaton and m is the number of transitions of the projected automaton.

Algorithm 1: Verification of weak k-step opacity

Input : A DES $G = (Q, \Sigma, \delta, I)$, $Q_S, Q_{NS} \subseteq Q$, $\Sigma_o \subseteq \Sigma$, and $k \in \mathbb{N}_\infty$.
Output: `true` iff G is weakly k-step opaque with respect to Q_S, Q_{NS}, and $P\colon \Sigma^* \to \Sigma_o^*$

1 Set $Y := \emptyset$
2 Compute the observer G^{obs} of G
3 Compute the projected automaton $P(G)$ of G
4 **for** *every reachable state X of G^{obs}* **do**
5 **for** *every state $x \in X \cap Q_S$* **do**
6 add state $(x, X \cap Q_{NS})$ to set Y
7 Construct H as the part of the full observer of G accessible from the states of the second components of Y
8 Compute the product automaton $\mathcal{C} = P(G) \times H$ with the states of Y as initial states
9 Use BFS to mark states of $\mathcal{C}$ reachable from Y in at most k steps
10 **if** *$\mathcal{C}$ contains a marked state of the form $(q, \emptyset)$* **then return** `false` **else return** `true`

The steps of the algorithm are quite intuitive, including the use of the Breadth-First Search (BFS) as described in detail in Cormen et al. (2009), which is used to search the underlying graph structure of the automaton $\mathcal{C}$ and to mark all its states that are reachable from an initial state in at most k steps. The correctness of the algorithm follows from the fact that the BFS search visits all nodes at distance d from the initial nodes before visiting any nodes at distance $d+1$, and hence the distance (aka the number of hops) is bounded by the number of states, and not by the parameter k.

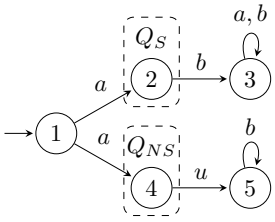

Fig. 1. A DES G.

However, the algorithm of Cormen et al. (2009) maintains an array to store the shortest distances of every node to an initial node. Since storing a number less than or equal to k requires $\log(k)$ bits, we need the space of size $O(\log(k)n2^n)$ to store the shortest distance of every state of $\mathcal{C}$ to an initial state of $\mathcal{C}$, because $\mathcal{C}$ has $O(n2^n)$ states.

In fact, we do not need to store the shortest distance of every state to an initial state of $\mathcal{C}$, but rather to keep track of the number of hops from the initial states made so far. We can achieve this by modifying the BFS of Cormen et al. (2009) so that we do not store the shortest distances for every state of $\mathcal{C}$, but only the current distance. We store the current distance in the queue used by the algorithm. In particular, after pushing all the initial states of $\mathcal{C}$ to the queue, we push number 0 in binary to the queue, representing that no hop has been done so far. When processing the initial states from the queue, number 0 separates the nonprocessed initial states of $\mathcal{C}$ from the noninitial states reachable in one hop from the processed initial states, which all appear in the queue after the number 0. After processing all the initial states, the BFS removes the separator 0 from the queue and pushes number 1 to the queue. At this point, the queue contains all noninitial states of $\mathcal{C}$ that are reachable from the initial states in one hop, followed by the separator 1, saying that the states of the queue are exactly those states that are at distance one from the states of the set Y and not less.

The algorithm proceeds this way, increasing the separator one by one every time the separator is processed, until it has either visited all the states of $\mathcal{C}$ or the separator stored in the queue is the number k in binary. All and only visited states of $\mathcal{C}$ are marked. Notice that this approach requires to store only one $\log(k)$-bit number rather than $n2^n$ such numbers.

To illustrate Algorithm 1, we consider weak one-step opacity of the DES G depicted in Fig. 1, where events a, b are observable, u is unobservable, state 2 is secret, and state 4 is nonsecret; the other states are neutral, meaning that they are neither secret nor nonsecret. The meaning of neutral states is not yet clear in the literature. They are fundamental in language-based opacity, but lose its meaning in state-based opacity. In any case, we cannot simply handle neutral states as nonsecret states.

The projected automaton $P(G)$ and the observer G^{obs} are depicted in Figs. 2 and 3, respectively. By the definition of the (full) observer, all the missing transitions in Fig. 3 indeed lead to state $\emptyset$, e.g., $\delta(\{1\}, b) = \delta(\{5\}, a) = \emptyset$. To keep the figures clear, we do not depict these transitions.

The only reachable state of G^{obs} that contains a secret state of G is the state $X = \{2, 4, 5\}$, and therefore we get

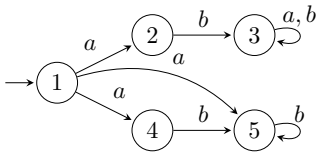

Fig. 2. The automaton $P(G)$.

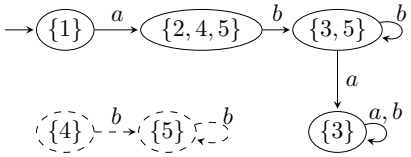

Fig. 3. The automaton G^{obs} (solid part) and the automaton H. The automaton H forming the relevant part of the full observer of G is obtained from G^{obs} by adding the dashed part.

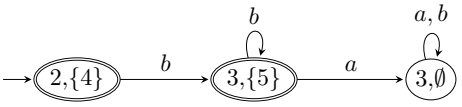

Fig. 4. The reachable part of $\mathcal{C}$.

that the set $Y = \{(2, \{4\})\}$. Notice that state $\{4\}$ is not in the observer G^{obs}, and therefore we need to add it to H together with all the states reachable from state $\{4\}$ in the full observer of G. That is, we construct the relevant part H of the full observer of G by extending G^{obs} by state $\{4\}$ and by all the reachable states from this state. The result (without the transitions to state $\emptyset$) is depicted in Fig. 3; both, the solid and the dashed part.

The marked part of $\mathcal{C} = P(G) \times H$ is depicted in Fig. 4. Since no state of the form $(\cdot, \emptyset)$ is marked in $\mathcal{C}$, the DES G is one-step opaque.

The following result formally justifies the correctness of Algorithm 1.

Theorem 2. A DES G is weakly k-step opaque with respect to Q_S, Q_{NS}, and P iff Algorithm 1 returns `true`. □

Considering the time and space complexity of Algorithm 1, we can show the following.

Theorem 3. The space and time complexity of Algorithm 1 is $O(n2^n)$ and $O((n+m)2^n)$, respectively, where n is the number of states of the input DES G and m is the number of transitions of $P(G)$. In particular, $m \leq \ell n^2$, where ℓ is the number of observable events. □

4. VERIFICATION OF STRONG K-STEP OPACITY

Weak k-step opacity could seem confidential enough. However, Falcone and Marchand (2014) pointed out that it is not as confidential as intuitively expected. In particular, the intruder may realize that the system previously was in a secret state, although it is not able to deduce the exact time when that happened, see Falcone and Marchand (2014) for more details and examples. As a result, they defined strong k-step opacity as a variation of k-step opacity with a higher level of confidentiality.

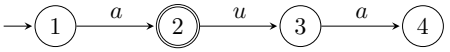

Fig. 5. A deterministic DES that is not confidential enough. The secret state is double circled.

Before we recall the definition of strong k-step opacity as formulated by Falcone and Marchand (2014), we illustrate the problem of weak k-step opacity in an example.

We consider the system depicted in Fig. 5, where state 2 is secret and the other states are nonsecret, and where the event a is observable and the event u is unobservable. Then, observing the sequence aa, the intruder realizes that the system previously was in the secret state 2, though it cannot say the exact time when that happened.

In the sequel, in accordance with the setting of Falcone and Marchand (2014), we consider strong k-step opacity only for deterministic DES where all states that are not secret are nonsecret, that is, $Q_{NS} = Q - Q_S$. That is, every state has its secret/nonsecret status and there are no neutral states.

Definition 4. Given a deterministic DES $G = (Q, \Sigma, \delta, q_0)$ and $k \in \mathbb{N}_\infty$. System G is *strongly k-step opaque (k-SSO)* with respect to the set Q_S of secret states and observation $P\colon \Sigma^* \to \Sigma_o^*$ if for every string $s \in L(G)$, there exists a string $w \in L(G)$ such that $P(s) = P(w)$ and for every prefix w' of w, if $|P(w)| - |P(w')| \leq k$, then $\delta(q_0, w') \notin Q_S$.

For an illustration, we again consider the system depicted in Fig. 5, where state 2 is secret and the event u is unobservable. The system is weakly one-step opaque, but not strongly one-step opaque, because, for $s = aua$, the only w with the same observation as s is $w = aua$, and hence the prefixes w' for which $|P(w)| - |P(w')| \leq 1$ are the strings $w' = a$, $w' = au$, and $w' = aua$. However, for $w' = a$, we obtain that $\delta(1, a) = 2 \in Q_S$, which violates the definition of strong one-step opacity.

It can be shown that the system of Fig. 5 is not strongly 0-step opaque, but it is weakly 0-step (current-state) opaque. Hence, the notions of strong 0-step opacity and weak 0-step (current-state) opacity do not coincide. It can further be shown that unobservable transitions from secret states to nonsecret states, as in our example, are the only issues making the difference between strong 0-step opacity and weak 0-step (current-state) opacity.

In the sequel, we consider only deterministic DESs where there are no unobservable transitions from secret states to nonsecret states. We call such systems *normal*.

We point out that every deterministic DES can be normalized (in linear time) in such a way that the normalization of a deterministic DES is a normal deterministic DES that preserves the property of being strongly k-step opaque. Therefore, considering, in the sequel, only normal deterministic DESs is without loss of generality.

We now show how to reduce strong k-step opacity to weak k-step opacity, which allows us to verify strong k-step opacity by checking weak k-step opacity.

Construction 5. Let $G = (Q, \Sigma, \delta, q_0)$ be a normal deterministic DES, $P\colon \Sigma^* \to \Sigma_o^*$ be the observation, and Q_S be the set of secret states. We construct a deterministic DES

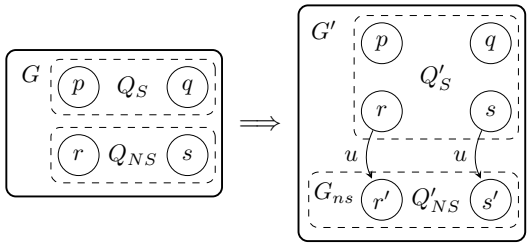

Fig. 6. An illustration of Construction 5 transforming strong k-step opacity to weak k-step opacity.

$$G' = (Q \cup Q'_{NS}, \Sigma \cup \{u\}, \delta', q_0)$$

as a disjoint union of G and $G_{ns} = (Q'_{NS}, \Sigma, \delta_{ns}, q'_0)$, where G_{ns} is obtained from G by removing all secret states and corresponding transitions, and $Q'_{NS} = \{q' \mid q \in Q_{NS}\}$ is a copy of Q_{NS} disjoint from Q. We use a new unobservable event u to connect G_{ns} to G so that we initialize $\delta' := \delta \cup \delta_{ns}$, and extend δ' by additional transitions (q, u, q') for every $q \in Q_{NS}$, cf. Fig. 6 for an illustration. The states of Q'_{NS} are the only nonsecret states of G', and hence the set of secret states of G' is the set $Q'_S = Q$. Finally, we define $P' \colon (\Sigma \cup \{u\})^* \to \Sigma_o^*$. ◇

The following theorem describes the relationship between strong k-step opacity and weak k-step opacity, and justifies the correctness of Algorithm 2.

Theorem 6. Let $G = (Q, \Sigma, \delta, q_0)$ be a normal deterministic DES, and let $G' = (Q \cup Q'_{NS}, \Sigma \cup \{u\}, \delta', q_0)$ be the DES obtained from G by Construction 5. Then, the DES G is strongly k-step opaque with respect to Q_S and P if and only if the DES G' is weakly k-step opaque with respect to Q'_S, Q'_{NS}, and P', where Q'_S, Q'_{NS}, and P' are as defined in Construction 5. □

Algorithm 2: Verification of strong k-step opacity

Input : A normal deterministic DES $G = (Q, \Sigma, \delta, q_0)$, $Q_S \subseteq Q$, $\Sigma_o \subseteq \Sigma$, and $k \in \mathbb{N}_\infty$.

Output: `true` iff G is strongly k-step opaque with respect to Q_S and $P \colon \Sigma^* \to \Sigma_o^*$

1 Transform G to G' by Construction 5
2 Use Algorithm 1 on G' with the set of secret states Q'_S, the set of nonsecret states Q'_{NS}, observable events Σ_o, and k
3 **return** the answer of Algorithm 1

We now illustrate the algorithm by an example. In particular, besides illustrating the algorithm, we also illustrate that the reasoning about strong k-step opacity is not only intuitively, but also practically more difficult than the reasoning about weak k-step opacity.

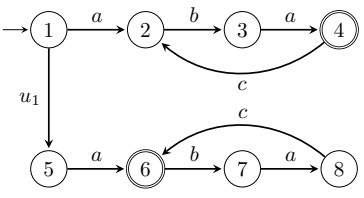

Fig. 7. A DES G.

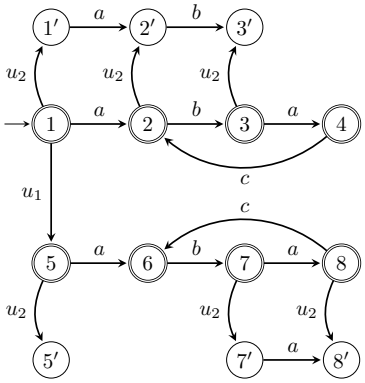

Fig. 8. The DES G' resulting from Construction 5 applied to G.

We adopt the DES G from Falcone and Marchand (2014) depicted in Fig. 7, where events a, b, c are observable, the event u_1 is unobservable, and states 4 and 6 are secret. Falcone and Marchand (2014) claim that G is strongly one-step opaque. Using our transformation to weak k-step opacity and our algorithm verifying weak k-step opacity, we show that it is not the case.

Since the DES G is normal, we proceed by the application of Construction 5, resulting in the DES G' depicted in Fig. 8. In particular, G' was constructed from G by adding one new unobservable event u_2 and six new nonsecret states, namely $Q'_{NS} = \{1', 2', 3', 5', 7', 8'\}$, and by making all the states 1 through 8 the secret states of G', that is, $Q'_S = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

Now, we apply Algorithm 1 to G' with the secret states Q'_S, the nonsecret states Q'_{NS}, observable events $\Sigma_o = \{a, b, c\}$, and $k = 1$. The observer G'^{obs} of G', and the automaton H are depicted (without the transitions to state $\emptyset$) in Fig. 9. The automaton $\mathcal{C} = P(G') \times H$ has eight initial states forming the set Y as depicted in Fig. 10. Since states $(2, \emptyset)$ and $(6, \emptyset)$ of $\mathcal{C}$ are reachable from states $(4, \{8'\}), (8, \{8'\}) \in Y$, respectively, in one step, G' is not weakly one-step opaque; and hence G is neither strongly one-step opaque. Indeed, looking at the DES G and observing the string $abac$, the intruder reveals that G is either in the secret state 6 at that instant, or must have been in the secret state 4 one step ago.

Algorithms verifying strong k-step opacity have been investigated in the literature. Falcone and Marchand (2014) have designed an algorithm based on the k-delay trajectory estimation. However, they have not analyzed the complexity of their algorithm. Recently, Ma et al. (2021) designed another algorithm with the complexity $O(\ell 2^{(k+2)n})$, where ℓ is the number of observable events and n is the number of states of the input DES. Even more recently, Wintenberg et al. (2022) discussed and experimentally compared algorithms based on the construction of a secret observer with complexity $O(\ell(k+3)^n)$, on the reverse comparison with complexity $O((n+m)(k+1)2^n)$, where $m \leq \ell n^2$, and on the construction of the k-delay trajectory estimator of Falcone and Marchand (2014), which they estimate to be of complexity $O(\ell(\ell+1)^k 2^n)$.

Analysing the complexity of Algorithm 2, it can be shown that, for a normal deterministic DES with n states, Con-

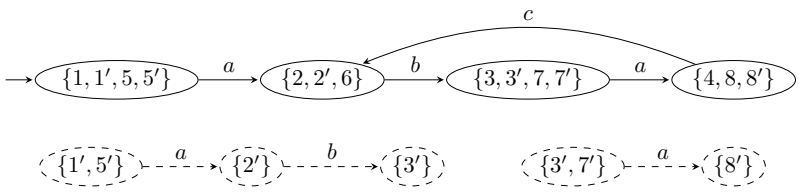

Fig. 9. The observer G'^{obs} of G', the solid part. The automaton H forming the relevant part of the full observer of G' is obtained from G'^{obs} by adding the dashed part.

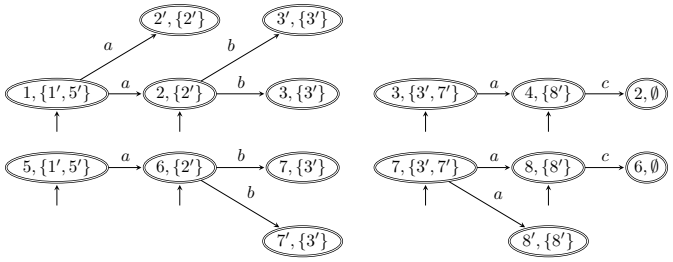

Fig. 10. The relevant part of $\mathcal{C}$, where the depicted states are reachable from the set Y in one step.

struction 5 results in a deterministic DES with at most $2n$ states, and that the observer of the resulting DES has at most 2^n states. Hence, the worst-case state complexity of Algorithm 2 is $O(n2^n)$, and the time complexity is thus $O((n+m)2^n)$. In any case, the complexity of Algorithm 2 does not depend on the parameter k.

5. CONCLUSIONS

We investigated and discussed the relation between the notions of weak and strong k-step opacity. We designed a new algorithm verifying weak k-step opacity that, compared with the existing algorithms, does not depend on k, and that has a lower complexity than the existing algorithms. Then, we discussed strong k-step opacity and transformed it to weak k-step opacity in linear time, obtaining thus a new algorithm to verify strong k-step opacity.

Finally, we point out that in the extended version (Balun and Masopust, 2022), we further discuss a modification of Algorithm 2 where the input is not restricted to normal deterministic DES. Namely, in the first step, the algorithm normalizes the input and then proceeds as Algorithm 2. Although the normalization produces a DES with up to twice more states compared with the input DES, we show that the worst-case complexity remains unchanged.

ACKNOWLEDGEMENTS

We gratefully acknowledge suggestions and comments of the anonymous referees.

REFERENCES

Badouel, E., Bednarczyk, M., Borzyszkowski, A., Caillaud, B., and Darondeau, P. (2007). Concurrent secrets. *Discrete Event Dynamic Systems*, 17, 425–446.

Balun, J. and Masopust, T. (2021). Comparing the notions of opacity for discrete-event systems. *Discrete Event Dynamic Systems*, 31, 553–582.

Balun, J. and Masopust, T. (2022). Verifying weak and strong k-step opacity in discrete-event systems. doi: 10.48550/arXiv.2204.01286. Preprint, extended version.

Bryans, J.W., Koutny, M., Mazaré, L., and Ryan, P.Y.A. (2008). Opacity generalised to transition systems. *International Journal of Information Security*, 7(6), 421–435.

Bryans, J.W., Koutny, M., and Ryan, P.Y. (2005). Modelling opacity using Petri nets. *Electronic Notes in Theoretical Computer Science*, 121, 101–115.

Cassandras, C.G. and Lafortune, S. (eds.) (2021). *Introduction to Discrete Event Systems*. Springer, third edition.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press.

Dubreil, J., Darondeau, P., and Marchand, H. (2008). Opacity enforcing control synthesis. In *WODES*, 28–35.

Falcone, Y. and Marchand, H. (2014). Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Systems*, 25, 531–570.

Hopcroft, J.E., Motwani, R., and Ullman, J.D. (2006). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Jacob, R., Lesage, J.J., and Faure, J.M. (2016). Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control*, 41, 135–146.

Jirásková, G. and Masopust, T. (2012). On a structural property in the state complexity of projected regular languages. *Theoretical Computer Science*, 449, 93–105.

Lan, H., Tong, Y., Guo, J., and Giua, A. (2020). Comments on "A new approach for the verification of infinite-step and K-step opacity using two-way observers" [Automatica 80 (2017) 162–171]. *Automatica*, 122, 109290.

Ma, Z., Yin, X., and Li, Z. (2021). Verification and enforcement of strong infinite- and k-step opacity using state recognizers. *Automatica*, 133, 109838.

Saboori, A. and Hadjicostis, C.N. (2011). Verification of K-step opacity and analysis of its complexity. *IEEE Transactions on Automation Science and Engineering*, 8(3), 549–559.

Saboori, A. and Hadjicostis, C.N. (2007). Notions of security and opacity in discrete event systems. In *IEEE CDC*, 5056–5061.

Saboori, A. and Hadjicostis, C.N. (2012). Verification of infinite-step opacity and complexity considerations. *IEEE Transactions on Automatic Control*, 57(5), 1265–1269.

Wintenberg, A., Blischke, M., Lafortune, S., and Ozay, N. (2022). A general language-based framework for specifying and verifying notions of opacity. *Discrete Event Dynamic Systems*, 32, 253–289.

Wong, K. (1998). On the complexity of projections of discrete-event systems. In *WODES*, 201–206.

Yin, X. and Lafortune, S. (2017). A new approach for the verification of infinite-step and K-step opacity using two-way observers. *Automatica*, 80, 162–171.