Transparência Baseada em Propriedade: uma Nova Noção de Utilidade para Sistemas a Eventos Discretos ⋆

Patricia Monica Campos-Mayer-Vicente, Felipe Gomes Cabral, Públio M. Lima, Marcos Vicente Moreira, Audine Subias, Yannick Pencolé

Transparência Baseada em Propriedade: uma Nova Noção de Utilidade para Sistemas a Eventos Discretos ⋆

Patrícia C. Mayer * **Felipe G. Cabral** * **Públio M. M. Lima** *
Marcos V. Moreira ** **Audine Subias** *** **Yannick Pencolé** ***

* *Departamento de Engenharia de Automação e Sistemas (EAS), Universidade Federal de Santa Catarina (UFSC), Campus Trindade, Florianópolis, 88.040-900, SC, Brasil (patricia.mayer@posgrad.ufsc.br, felipe.gomes.cabral@ufsc.br, publio.lima@ufsc.br).*
** *COPPE - Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Cidade Universitária, Ilha do Fundão, Rio de Janeiro, 21.945-970, RJ, Brasil (moreira.mv@poli.ufrj.br)*
*** *LAAS-CNRS, CNRS, University of Toulouse, France, (subias@laas.fr, yannick.pencole@laas.fr)*

Abstract: The property related to the security of Cyber-Physical Systems (CPSs) is called opacity, which aims to hide sensitive information from an agent who gains unauthorized access to the industrial communication network. A system is said to be opaque when sensitive information is preserved in a passive cyber attack. However, this information will be hidden from legitimate receivers, such as a SCADA application in an opaque system. Therefore, the availability of system information, known as utility, to trustworthy agents is fundamental to operating such applications efficiently, which establishes a trade-off between the privacy of information against unauthorized agents and the data to legitimate receivers. In this paper, a new notion of utility, called Property-Based Transparency (PBT), is defined. A system is said to be transparent if legitimate receivers can determine whether a given property is satisfied in the system before that property becomes false. A method for verifying PBT is presented, while the boundaries between PBT and opacity are discussed. In addition, a case study is presented to show that, in some cases, the same information can be transparent to legitimate receivers and opaque to unauthorized agents.

Resumo: A propriedade relacionada à segurança de sistemas ciberfísicos que tem como objetivo esconder informações sensíveis de uma rede de comunicação industrial de um agente não autorizado é chamada opacidade. Um sistema é considerado opaco quando informações sigilosas são preservadas em uma situação de ataque cibernético passivo. Entretanto, em um sistema opaco, essas mesmas informações sensíveis não estarão disponíveis para receptores legítimos, como, por exemplo, em uma aplicação SCADA. Dessa forma, neste artigo, uma nova noção de utilidade, chamada de transparência baseada em propriedade (TBP), é definida. Um sistema é dito ser TBP se os receptores legítimos conseguem discernir se uma determinada propriedade é válida no sistema antes que essa propriedade se torne falsa. Um método para verificação da TBP é apresentado, e uma discussão sobre os limites entre TBP e a opacidade é feita. Além disso, um estudo de caso é apresentado para mostrar que, em alguns casos, é possível que a mesma informação seja transparente para receptores legítimos e opaca para agentes não autorizados.

Keywords: Cybersecurity; Utility; Transparency; Discrete Event Systems; Automation.
Palavras-chaves: Cibersegurança; Utilidade; Transparência; Sistemas a Eventos Discretos; Automação.

1. INTRODUÇÃO

Sistemas Ciber-Físicos (SCFs) possuem três camadas de abstração intrinsecamente conectadas: (*i*) virtual; (*ii*) física; e (*iii*) a rede de comunicação que interconecta as camadas anteriores. Essas camadas interagem constantemente para operar um processo com segurança e eficiência a partir da troca de informações entre os seus dispositivos, elemento fundamental para a Indústria 4.0 (Ding et al., 2018). Embora essa integração tenha aumentado a segurança operacional desses sistemas, as redes que comunicam dados confidenciais são suscetíveis a ataques cibernéticos.

Esses ataques podem ser realizados para obter informações industriais secretas ou danificar processos, operadores, instalações ou até mesmo a população. Usualmente, essas informações também são comunicadas a um receptor que, ao interpretar com precisão os dados transmitidos, pode intervir nas tomadas de decisão. Além disso, três elementos garantem a segurança dos CPSs: disponibilidade, integridade e confidencialidade (Alguliyev et al., 2018). No contexto de Sistemas a Eventos Discretos (SEDs), o problema para garantir que as informações úteis da rede do sistema sejam obtidas pelo receptor é denominada utilidade (Dwork, 2006; Wu et al., 2018).

O problema de utilidade pode ser considerado como "o outro lado da moeda" a partir do problema de opacidade. Um sistema é opaco quando as informações secretas não são reveladas a um observador externo não-autorizado, que geralmente acessa o sistema por meio de um ataque de espionagem (Fritz et al., 2019; Lima et al., 2022). O atacante atua passivamente para obter os dados transmitidos na rede de comunicação entre a planta e o receptor legítimo, sem modificar o comportamento do sistema.

A propriedade de opacidade garante que a confidencialidade do sistema não seja violada, enquanto a utilidade assegura que um receptor obtenha informações úteis sobre o sistema e a disponibilidade dos componentes durante os processos. Diversos trabalhos na literatura definem diferentes noções de opacidade, como a *Current-State Opacity* (Saboori and Hadjicostis, 2007), *K-step Opacity* (Saboori and Hadjicostis, 2007), e *Initial-State Opacity* (Saboori and N. Hadjicostis, 2013), entre outros (Jacob et al., 2016; Oliveira et al., 2023), por meio de distintas perspectivas (Lin, 2011). Além disso, técnicas para garantir a opacidade em sistemas não-opacos também foram propostas (Yiding et al., 2018; Li et al., 2023).

Embora as técnicas para garantir a opacidade aumentem a privacidade do sistema a partir da perspectiva de um intruso, informações úteis podem não ser transmitidas para um receptor legítimo. Portanto, o principal problema na comunidade de SEDs relacionado à opacidade e suas estratégias de aplicação é que, sempre que essa propriedade é satisfeita, o comportamento do sistema é opaco tanto para um intruso quanto para um receptor legítimo (Barcelos and Basilio, 2023). Com o intuito de evitar essa desvantagem, alguns trabalhos introduziram noções de utilidade (Dwork, 2006) que devem ser preservadas quando uma estratégia de aplicação de opacidade é implementada (Wu et al., 2018; Wintenberg et al., 2022; Cardoso et al., 2023; Barcelos and Basilio, 2023).

Em Wu et al. (2018), a noção de utilidade foi introduzida no contexto do SEDs, em que a propriedade de utilidade é satisfeita quando o número de transições entre o estado atual do sistema e o estado estimado pelo receptor é menor do que um determinado número. No entanto, em Wu et al. (2018), os estados secretos não podem ser considerados estados úteis, pois nunca devem ser reportados. A mesma noção de utilidade é explorada em Winterberg et al. (2022) e Cardoso et al. (2023). Recentemente, uma nova noção de utilidade foi proposta em Barcelos and Basilio (2023). Um sistema é *"utility-ensured"* quando o receptor legítimo pode determinar com precisão quando a planta atinge cada um dos estados úteis. Além disso, o estado útil estimado também deve corresponder ao estado atual da planta. Portanto, se o receptor alcança uma estimativa composta apenas de estados úteis, o sistema não será *"utility-ensured"*, pois o receptor não consegue diferenciar exclusivamente todos os estados úteis.

Neste trabalho, o cenário no qual o receptor necessita indicar quando uma determinada propriedade de interesse é satisfeita antes que o sistema atinja um estado em que essa propriedade não seja mais verdadeira é abordado. Esse problema é motivado pelo cenário em que um supervisor descentralizado requer informações sobre, por exemplo, a localização geográfica de um agente ou alertar um operador sobre a ocorrência de uma falha. Nesse contexto, o receptor deve detectar que um estado útil foi alcançado antes que o sistema evolua para um estado não útil.

Além disso, esse problema não pode ser abordado com a noção de utilidade introduzida em Wu et al. (2018), visto que o foco deste trabalho é detectar se o sistema atingiu uma determinada região útil e não se uma distância mínima de um conjunto de estados para o atual foi preservada. Este trabalho também se diferencia da noção de utilidade proposta em Barcelos and Basilio (2023), pois não é necessário que o receptor consiga distinguir cada estado útil, ou seja, o receptor deve sempre detectar somente se uma propriedade de interesse foi satisfeita.

Nesse contexto, uma nova noção de utilidade, denominada Transparência Baseada em Propriedade (TBP), cuja propriedade de interesse é a informação útil que o receptor precisa saber, é proposta. Um método para verificar se um sistema é transparente com base em propriedades é apresentado. Além disso, a noção de TBP é explorada no contexto de opacidade aplicada ao diagnóstico de falhas.

Ao abordar a opacidade, considera-se que o atacante observa os eventos transmitidos no canal de comunicação e não possui certeza do estado atual do sistema no início do ataque. Portanto, a partir dessa hipótese, é possível mostrar que um sistema pode ser *"current-state opaque"* (Saboori and Hadjicostis, 2007) e TBP, mesmo quando um estado ou conjunto de estados é secreto para o atacante e útil para o receptor legítimo. Além disso, é possível considerar a detecção da ocorrência de um evento de falha (Sampath et al., 1996) como a propriedade de interesse, implicando a TBP para o receptor. Exemplos e um estudo de caso são apresentados ao longo do texto para ilustrar os resultados e as comparações.

⋆ Este trabalho foi parcialmente financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Código de Financiamento 001 e FAPESC.

Este artigo está organizado da seguinte forma: na seção 2 a notação utilizada neste trabalho e fundamentação teórica são apresentados. Na seção 3, o problema de utilidade em Sistemas a Eventos Discretos é ilustrado e a noção de Transparência Baseada em Propriedade é introduzida. Na seção 4, um método para verificação da TBP é proposto. Na seção 5, um estudo de caso é utilizado para ilustrar a aplicação da noção de TBP no contextos de opacidade e diagnóstico de falhas. As conclusões são apresentadas na seção 6.

2. FUNDAMENTAÇÃO TEÓRICA

Um autômato é denotado por $G = (Q, \Sigma, f, q_0)$, em que Q é o conjunto finito de estados, Σ é o conjunto finito de eventos, $f : Q \times \Sigma \to Q$ é a função determinística de transição, e q_0 indica o estado inicial. A função de eventos ativas de G é denotada por: $\Gamma_G : Q \to 2^{\Sigma}$, em que $\Gamma_G(q) = \{\sigma \in \Sigma : f(q, \sigma)!\}$, em que ! representa que $f(q_0, \sigma)$ é definida. O domínio da função de transição pode ser estendida para $Q \times \Sigma^*$, em que Σ^* denota o fecho de Kleene de Σ, como $f(q, \varepsilon) = q$ e $f(q, s\sigma) = f(f(q, s), \sigma)$, para todo $s \in \Sigma^*$ e $\sigma \in \Sigma$, em que ε representa a sequência vazia. A linguagem gerada por G é definida como $L = \{s \in \Sigma^* : f(q_0, s)!\}$. Um caminho p de comprimento k de G é definido como $p = (q_1, \sigma_1, q_2, \ldots, \sigma_{k-1}, q_k)$, em que $q_i \in Q$, para $i = 1, \ldots, k$, $\sigma_i \in \Sigma$, para $i = 1, \ldots, k-1$, e $f(q_i, \sigma_i) = q_{i+1}$, for $i = 1, \ldots, k-1$. Portanto, existe uma sequência de eventos $s = \sigma_1\sigma_2 \ldots \sigma_{k-1}$ associada a cada caminho p de comprimento k, tal que $f(q_1, s)$ é definida. O comprimento da sequência $s \in \Sigma^*$ é definida como $\|s\|$. O prefixo-fechamento de uma linguagem L é definida como $\bar{L} = \{s \in \Sigma^* : (\exists t \in \Sigma^*)[st \in L]\}$. Sejam G_1 e G_2 dois autômatos, a composição paralela desses autômatos é denotada como $G = G_1 \| G_2$ (Cassandras and Lafortune, 2008).

O conjunto de eventos de G pode ser particionado como $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, em que Σ_o e Σ_{uo} representam o conjunto de eventos observáveis e não-observáveis, respectivamente. A operação de projeção é denotada por $P_o : \Sigma^* \to \Sigma_o^*$, em que $P_o(\varepsilon) = \varepsilon$, $P_o(\sigma) = \sigma$, se $\sigma \in \Sigma_o$, e $P_o(\sigma) = \varepsilon$, se $\sigma \in \Sigma \setminus \Sigma_o$, e $P_o(s\sigma) = P_o(s)P_o(\sigma)$, para todo $s \in \Sigma^*$ e $\sigma \in \Sigma$. Portanto, $P_o(s)$ representa a observação de s. O autômato que possui a linguagem igual a $P_o(L)$ é denominado como observador de G, denotado como $Obs(G, q_0)$ (Cassandras and Lafortune, 2008). Cada estado de $Obs(G, q_0)$ corresponde a estimativa de estado de G considerando o conjunto de eventos observáveis, Σ_o.

A noção de utilidade tem sido amplamente explorada na literatura desde o artigo seminal de Wu et al. (2018), e diferentes noções foram propostas. Na sequência, uma das definições de utilidade, introduzida em Barcelos and Basilio (2023), é apresentada.

Definição 1. (Utility-Ensured Systems). Dado um sistema $G = (Q, \Sigma, f, q_0)$, a projeção $P_o : \Sigma^* \to \Sigma_o^*$, e o conjunto de estados úteis $Q_U \subseteq Q$. Portanto, o sistema G é *"utility-ensured"* com relação a P_o e Q_U, se $\forall q_U \in Q_U$ e $\forall s \in L$ tal que $f(q_0, s) = q_U$, $\forall q' \in Q \setminus \{q_U\}$ e $\forall s' \in L$ tal que $f(q_0, s') = q'$, $P_o(s) \neq P_o(s')$. □

É importante ressaltar que, conforme a Definição 1, G é *"utility-ensured"* se um estado útil for sempre estimado exclusivamente sob a projeção P_o. Além disso, alguns estados nem sempre podem ser determinados de forma exclusiva em uma aplicação real devido à ocorrência de eventos não-observáveis. Por exemplo, para um determinado sistema em que uma transição de um estado q para outro estado q' é rotulada com um evento não-observável, ou seja, $f(q, \sigma_{uo}) = q'$, em que $\sigma_{uo} \in \Sigma_{uo}$, e $q, q' \in Q_U$, é impossível ter uma estimativa com apenas o estado q, e G não é trivialmente *"utility-ensured"*.

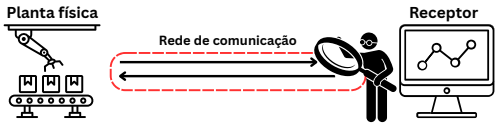

Figura 1. Representação da troca de informações entre a planta e o receptor.

3. FORMULAÇÃO DO PROBLEMA

Neste trabalho, é considerado que a planta e seu supervisor (também chamado de receptor), por exemplo, um sistema SCADA, podem trocar informações com base em ocorrências de eventos observáveis, conforme ilustrado na Figura 1. As informações transmitidas são utilizadas pelo receptor para estimar o comportamento da planta. Por exemplo, para determinar o próximo evento de controle para o sistema seguir as especificações projetadas ou detectar um comportamento não desejado que deve disparar um alarme ao operador. Portanto, a arquitetura de comunicação é projetada para garantir que o receptor possa distinguir se e quando a planta está em um dos estados de um determinado conjunto, chamado de estados úteis.

Em diversas aplicações práticas, o receptor necessita saber em que momento uma determinada propriedade $\mathcal{P}$, associada a um conjunto de estados, é satisfeita antes que o sistema evolua para um estado em que $\mathcal{P}$ seja falso. Por exemplo, no controle coordenado de múltiplos robôs para o cenário de busca e resgate urbano considerado em Simon and Baldissera (2023), o supervisor deve saber se um determinado robô está em uma região geográfica para emitir um evento de controle, como a busca de vítimas. Nesse caso, a propriedade $\mathcal{P}$ é verdadeira enquanto o robô estiver na região de interesse, e o supervisor não precisa saber precisamente a localização do robô durante sua missão. O problema de garantia da opacidade associada a ocorrência de uma falha é outro contexto em que o supervisor está interessado em detectar quando uma determinada propriedade $\mathcal{P}$ é verdadeira, em vez de estimar e diferenciar cada estado após a falha. Neste trabalho, a noção de utilidade proposta, denominada transparência, é baseada em uma propriedade de interesse relacionada às informações conhecidas e necessárias para o receptor legítimo alcançar um objetivo desejado. Em seguida, a noção de transparência é formalmente introduzida.

3.1 Noção de transparência

Para introduzir a noção de transparência baseada em propriedade, é necessário definir as notações a seguir. O conjunto de estados úteis Q_U, composto por estados que satisfazem a propriedade $\mathcal{P}$, é definido como $Q_U = \{q \in Q : q \vDash \mathcal{P}\}$. O conjunto de estados não-úteis é dado

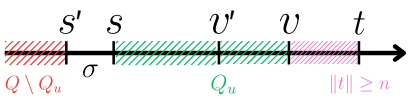

Figura 2. Interpretação gráfica da Definição 2.

por $Q_{\neg U} = Q \setminus Q_U$. A linguagem que contém todas as sequências de eventos da linguagem L que alcançam um estado não-útil é definida como $L_{\neg U}$, ou seja, um estado $q \in Q_{\neg U}$, em que $L_{\neg U} = \{s \in L : f(q_0, s) \notin Q_U\}$. A noção de TBP é formalmente definida na sequência.

Definição 2. (Transparência). Dado um sistema $G = (Q, \Sigma, f, q_0)$, a projeção $P_o : \Sigma^* \to \Sigma_o^*$, e o conjunto Q_U, G é considerado TBP com relação a P_o e Q_U, se $\forall s = s'\sigma \in L, \sigma \in \Sigma$, tal que $f(q_0, s) \in Q_U$ e $f(q_0, s') \in Q \setminus Q_U$, ou $s = \varepsilon$ e $q_0 \in Q_U$:

$$(\exists n \in \mathbb{N})(\forall t \in L/s : \|t\| \geq n)$$
$$(\exists v \in \overline{\{t\}} : \forall v' \in \overline{\{v\}}, f(q_0, sv') \in Q_U)$$
$$(P_o(sv) \neq P_o(\omega), \forall \omega : f(q_0, \omega) \notin Q_U)$$

□

Na Definição 2, o sistema G é TBP de acordo com uma propriedade específica se para todas as sequências $s \in L$ que alcançam um estado em Q_U, de modo que $s = s'\sigma$ e $f(q_0, s') \in Q_{\neg U}$, s possui ao menos um sufixo v que não leva o sistema a um estado em $Q_{\neg U}$ de modo que $P_o(sv) \neq P_o(\omega)$, para todos os $\omega \in L_{\neg U}$. A definição de TBP é ilustrada graficamente na Figura 2. Note que antes da ocorrência do evento $\sigma \in \Sigma$, o sistema está em uma região que não pertence a Q_U, em que $f(q_0, s') \in Q_{\neg U}$. Após a ocorrência de σ, em que $f(q_0, s'\sigma) = s \in Q_U$, deve existir uma sequência v após s, os quais todos os seus prefixos pertencem a Q_U. Além disso, não é possível determinar que as sequências com comprimento arbitrariamente longo após v, ou seja, os prefixos de t após v pertencem à região útil. No entanto, visto que o objetivo do supervisor é determinar se a propriedade $\mathcal{P}$ foi satisfeita, enquanto $P_o(sv) \neq P_o(\omega), \forall \omega \in L_{\neg U}$, a TBP também é satisfeita.

Em outras palavras, conforme a Definição 2, um sistema G é TBP se, para todas as sequências $s = s'\sigma$ que alcançam um estado $q \in Q_U$, em $f(q_0, s') \in Q_{\neg U}$, o receptor, com base na projeção natural, pode determinar com certeza que a propriedade $\mathcal{P}$ é satisfeita antes que G alcance um estado que não pertença a Q_U. Note que todas as sequências s que alcançam um estado em Q_U estão configurados como $s = s'\sigma$, $\sigma \in \Sigma$, exceto quando o estado inicial $q_0 \in Q_U$. Nesse caso, é necessário considerar também a sequência $s = \varepsilon$. A seguir, um exemplo de um sistema TBP conforme a Definição 2 é apresentado.

Exemplo 1. Considere o autômato G ilustrado na Figura 3, em que $\Sigma_o = \{a, b\}$ e $\Sigma_{uo} = \{\sigma_u\}$. Considere que esse sistema transmite os eventos a e b para o receptor, que deve identificar quando a propriedade $\mathcal{P}$ é satisfeita. Nesse sistema, os estados $\{0\}$, $\{3\}$, e $\{5\}$ satisfazem $\mathcal{P}$ e, portanto, $Q_U = \{0, 3, 5\}$. O sistema G é transparente conforme a Definição 2 se, após alcançar um estado em $q_U \in Q_U$, o receptor legítimo estima um subconjunto dos estados de Q_U antes de G alcançar um estado $q \in Q_{\neg U}$. Ao analisar o observador de G, $Obs(G, q_0)$, ilustrado na Figura 4, é possível constatar que no momento em

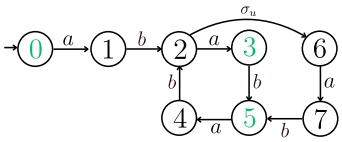

Figura 3. Autômato G do Exemplo 1, para $Q_U = \{0, 3, 5\}$.

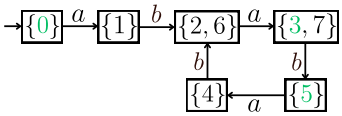

Figura 4. $Obs(G, q_0)$ do Exemplo 1, para $Q_U = \{0, 3, 5\}$.

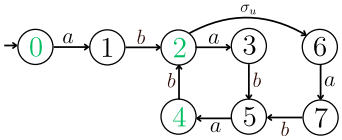

Figura 5. Autômato G do Exemplo 1, para $Q_U = \{0, 2, 4\}$.

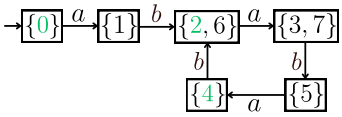

Figura 6. $Obs(G, q_0)$ do Exemplo 1, para $Q_U = \{0, 2, 4\}$.

que o sistema alcança os estados $\{0\}$ e $\{5\}$, o receptor tem certeza que $\mathcal{P}$ foi satisfeita. Note que quando o sistema alcança o estado $\{3\}$, o receptor não é capaz de identificar se o sistema está no estado $\{3\} \in Q_U$ ou $\{7\} \in Q_{\neg U}$, de acordo com $Obs(G, q_0)$. Entretanto, quando a estimativa de estado do receptor corresponde a $\{3, 7\}$, o único evento viável é b, em que $Obs(G, q_0)$ alcança o estado $\{5\}$, uma estimação de estados cujo $\mathcal{P}$ é satisfeita. Portanto, nesse exemplo, quando a estimativa de estado corresponde a $\{3, 7\}$, existem somente duas possibilidades após a observação do evento b: (i) o sistema estava no estado $\{3\}$ e alcançou o estado $\{5\}$, garantindo que o receptor está seguro que a propriedade $\mathcal{P}$ foi satisfeita no estado $\{3\}$ e continua válida após alcançar o estado $\{5\}$; ou (ii) o sistema estava no estado $\{7\}$ e alcançou o estado $\{5\}$, o que satisfaz $\mathcal{P}$. Portanto, G é TBP em relação a P_o e $Q_U = \{0, 3, 5\}$. Considere que agora, para o mesmo sistema, os estados que satisfazem a propriedade $\mathcal{P}$ são $\{0\}$, $\{2\}$, e $\{4\}$, apresentado na Figura 5. Sendo assim, nesse caso, $Q_U = \{0, 2, 4\}$. Ao analisar o observador de G, $Obs(G, q_0)$, ilustrado na Figura 6, é possível verificar que o receptor pode determinar que $\mathcal{P}$ foi satisfeita quando o sistema alcança o estado $\{0\}$, ou seja, $s = \varepsilon$. Em seguida, após a observação da sequência $s = ab$, o receptor não consegue determinar se o sistema está no estado $\{2\} \in Q_U$ ou $\{6\} \in Q_{\neg U}$. Conforme a Definição 2, o receptor deve estimar um subconjunto dos estados de Q_U antes de G alcançar um estado $q \in Q_{\neg U}$. Entretanto, o único evento viável a partir dos estados $\{2\}$ ou $\{6\}$ é a, cuja estimativa de estado corresponde a $\{3, 7\} \in Q_{\neg U}$. Portanto, o receptor não consegue determinar que a propriedade $\mathcal{P}$ foi satisfeita antes de alcançar uma estimativa apenas com estados não úteis. Sendo assim, G não é TBP em relação a P_o e $Q_U = \{0, 2, 4\}$.

□

É importante ressaltar que o sistema ilustrado na Figura 3 não é *utility-ensured* segundo a Definição 1, proposta em Barcelos and Basilio (2023), pois quando o sistema alcança o estado $\{3\}$, o receptor não consegue estimar unicamente esse estado em G.

Na sequência, o algoritmo para verificação da TBP em um sistema G é apresentado.

4. VERIFICAÇÃO DA TBP

A verificação da TBP de um sistema G é dado pelo algoritmo 1. Note que como $V_T = (Q_V, \Sigma, f_V, q_{0,V})$ é obtido pela composição paralela entre a planta G e seu observador G_{obs}, V_T mapeia todas as sequências de eventos que possuem a mesma projeção mantendo o estado exato alcançado por cada sequência de L em sua primeira coordenada. Além disso, cada estado $q_V \in Q_V$ tem a configuração $q_V = (q, q_{obs})$, em que $q \in Q$ e $q_{obs} \in Q_{obs}$. Portanto, cada estado $q_V \in Q_V$ de V_T possui duas coordenadas, em que, para todas as sequências $s \in L$, a primeira coordenada representa o estado q de G alcançado após s, $q = f(q_0, s)$, e a segunda indica a estimativa de estado q_{obs} após a observação $P_o(s)$. Na sequência, um exemplo para ilustrar o uso do algoritmo 1 para verificar a TBP é apresentado.

Algoritmo 1 Verificação - TBP

1: Calcule o observador de G, $G_{obs} = Obs(G, q_0)$.
2: Calcule o autômato verificador $V_T = (Q_V, \Sigma, f_V, q_{0,V})$, em que $V_T = G \| G_{obs}$.
3: Identifique todas as sequências em que $f_V((q, q_{obs}), \sigma) = q'_V$, tal que $q'_V = (q', q'_{obs}) \in Q_V$, $q \in Q_{\neg U}$ e $q' \in Q_U$.
4: Verifique a existência de ciclos $p = (q_{V_1}, \sigma_1, \ldots, q_{V_k}, \sigma_k, q_{V_1})$ nas sequências identificadas, tal que a primeira coordenada de todos os estados de p pertencem à Q_U e a segunda coordenada de todos os estados de p, possui estados que pertencem tanto a Q_U quanto a $Q_{\neg U}$. Em caso afirmativo, retorne que G não é TBP.
5: Verifique se a partir do estado atual de V_T, os eventos viáveis em que $q_V \in Q_U$, $\Gamma_G(q_V)$, alcançam um estado não-útil, $q'_V \in Q_{\neg U}$. Em caso afirmativo, retorne que G não é TBP.
6: Em caso negativo nos passos 4, 5 e 6, retorne que G é TBP.

Exemplo 2. Considere o autômato G apresentado na Figura 3, em que $\Sigma_o = \{a, b\}$ e $Q_U = \{0, 3, 5\}$. O autômato verificador de G, V_T, calculado a partir do algoritmo 1, é ilustrado na Figura 7. Ao observar a sequência $s = \varepsilon$, o receptor consegue identificar que o sistema alcançou o estado $\{0\}$. Note que para todas as sequências em que $f_V((q, q_{obs}), \sigma) = q'_V$, tal que $q'_V = (q', q'_{obs}) \in Q_V$, $q \in Q_{\neg U}$ e $q' \in Q_U$, não existem ciclos de acordo com as características apresentadas no passo 4 do algoritmo 1. Além disso, quando o verificador alcança o estado $q_V = 3, \{3, 7\}$, o único evento viável é b, tal que $f(q, b) = 5 \in Q_U$. Portanto, o passo 5 do algoritmo 1 não é satisfeito e o sistema G é TBP. □

5. ESTUDO DE CASO

Nesta seção, um estudo de caso para ilustrar a aplicação da TBP no contexto de opacidade, especificamente a

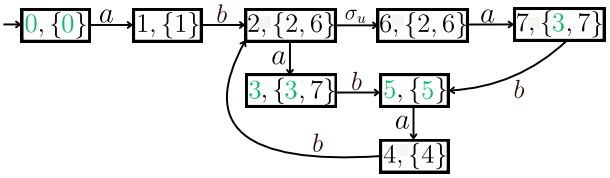

Figura 7. Autômato verificador V_T do exemplo 2.

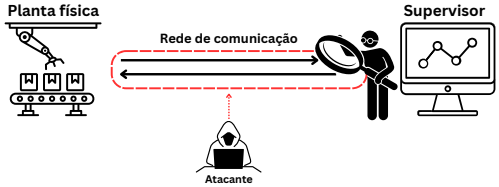

Figura 8. Representação de um ataque passivo na rede de comunicação entre a planta e o supervisor.

"Current-State Opacity" (Saboori and Hadjicostis, 2007), associada ao domínio de diagnóstico de falhas (Sampath et al., 1996) é apresentado. Considere que o canal de comunicação entre a planta e o supervisor possa ser espionado por um agente não autorizado, como ilustrado na Figura 8. Nesse contexto, presume-se que o atacante possui conhecimento do modelo completo do sistema e pode observar todas as ocorrências de eventos após o início do ataque. Além disso, considera-se que o atacante não está sincronizado com a planta, ou seja, sua primeira observação pode ser realizada a qualquer momento após a inicialização do sistema. Portanto, assim que o ataque é iniciado, o atacante deseja estimar o estado atual do sistema com base em suas observações (Saboori and Hadjicostis, 2007). Além disso, considera-se que o receptor legítimo inicia sua observação sincronicamente com o início do funcionamento do sistema, ou seja, observa o comportamento da planta a partir do seu estado inicial.

Suponha que tanto o atacante quanto o supervisor estejam interessados em identificar a ocorrência de uma falha durante os processos de um determinado sistema. Nesse caso, para garantir a disponibilidade do sistema em caso de um comportamento indesejado, a ocorrência da falha é considerada a propriedade de interesse para o receptor legítimo. Entretanto, a confidencialidade do sistema deve ser preservada para evitar possíveis intervenções e danos provocados pelo vazamento de dados secretos. Portanto, a ocorrência da falha deve ser a propriedade *transparente* para o supervisor e *opaca* para o atacante.

Nesse contexto, deseja-se verificar se o sistema é TBP e *"current-state opaque"*, conforme a definição apresentada a seguir (Saboori and Hadjicostis, 2007; Jacob et al., 2016; Lafortune et al., 2018).

Definição 3. (Current-State Opacity - CSO). Dado um sistema $G = (Q, \Sigma, f, Q_0)$, a projeção natural P_o, e um conjunto de estados secretos $Q_S \subseteq Q$, G é *"current-state opaque"* se:

$$\forall q_i \in Q_0 \text{ e } \forall s \in L(G, q_i) \text{ tal que } f(q_i, s) \in Q_S,$$
$$\exists q_j \in Q_0 \text{ e } \exists s' \in L(G, q_j) \text{ tal que}$$
$$f(q_j, s') \in Q \setminus Q_S \text{ e } P_o(s) = P_o(s').$$

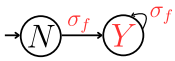

Figura 9. Autômato rotulador A_ℓ.

□

Note que na Definição 3, o estado inicial de G pode ser um conjunto de estado, inclusive igual a $Q_0 = Q$. Em Saboori (2011), um método para verificação da CSO é apresentado, o qual se baseia em um autômato estimador de estados $\mathcal{E} = Obs(G, Q)$, em que Q representa o conjunto de estados iniciais. Um sistema G é considerado CSO se nenhuma estimativa de estados de $\mathcal{E}$ é um subconjunto dos estados secretos Q_S.

No contexto de diagnóstico de falhas, o supervisor está interessado em identificar a ocorrência de um evento de falha não-observável com base em suas observações do comportamento do sistema. Portanto, a propriedade $\mathcal{P}$ é verdadeira para todos os estados alcançados após o evento de falha. Visto que pode existir em G estados alcançados após a ocorrência de um evento de falha e também por uma sequência de eventos livres de falha, a linguagem de G deve ser diagnosticável. A definição de diagnosticabilidade de SEDs é formalmente apresentada a seguir (Sampath et al., 1996), em que L_N é a linguagem livre de falha do sistema e o conjunto de eventos de falha é denotado por Σ_f, em que $\Sigma_f \subseteq \Sigma_{uo}$.

Definição 4. A linguagem gerada por G, L, é diagnosticável em relação à projeção $P_o : \Sigma^\star \to \Sigma_o^\star$ e Σ_f se

$$(\exists z \in \mathbb{N})(\forall s \in L \setminus L_N)(\forall st \in L \setminus L_N)$$
$$(\|t\| \geq z \Rightarrow P_o(st) \notin P_o(L_N))$$

□

Além disso, o autômato rotulador A_ℓ apresentado na Figura 9 é utilizado para distinguir os estados de acordo com as sequências executadas pelo sistema. A composição paralela entre a planta e o autômato A_ℓ, resulta em $G_\ell = (Q_\ell, \Sigma, f_\ell, q_{0,\ell}) = G \| A_\ell$. Os estados de G_ℓ são rotulados com N se o estado for alcançado após a ocorrência de uma sequência de eventos livre de falha e Y se o estado for alcançado após a ocorrência de um evento de falha (Cassandras and Lafortune, 2008). Sendo assim, o conjunto de estados úteis é definido como $Q_U = \{q_\ell \in Q_\ell : q_\ell = (q, Y)\}$.

Considere o autômato ilustrado na Figura 10. Ao realizar a composição paralela entre G e A_ℓ, o autômato G_ℓ, ilustrado na Figura 11, é obtido. Nesse caso, o conjunto de estados úteis é igual a $Q_U = \{1Y; 3Y\}$, e o observador de G_ℓ, $Obs(G_\ell)$, é apresentado na Figura 12. Note que conforme a estimativa de estados de $Obs(G_\ell)$, o receptor legítimo identifica a ocorrência do evento de falha após a observação do evento b. Sendo assim, G é diagnosticável em relação a Σ_f e P_o, o que implica que G é TBP em relação a Q_U e P_o.

Em seguida, para determinar se a ocorrência do evento de falha é *opaca* a partir da perspectiva do atacante, o estimador $\mathcal{E} = Obs(G_\ell, Q_\ell)$, ilustrado na Figura 13, é calculado. Note que não existe um estado em $\mathcal{E} = Obs(G_\ell, Q_\ell)$ que seja um subconjunto de Q_U. Portanto, a detecção da ocorrência do evento de falha é opaca para o atacante.

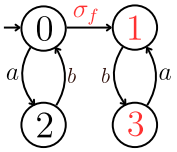

Figura 10. Autômato G com o comportamento pós-falha.

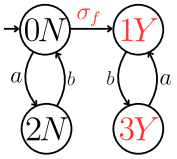

Figura 11. Autômato $G_\ell = G \| A_\ell$.

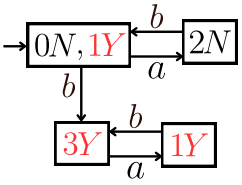

Figura 12. Observador a partir da perspectiva do supervisor, $Obs(G_\ell)$.

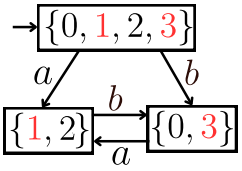

Figura 13. Observador a partir da perspectiva do atacante, $Obs(G_\ell, Q_\ell)$.

6. CONCLUSÃO

Neste trabalho, uma nova noção de utilidade chamada Transparência Baseada em Propriedade é proposta. A TBP se baseia na capacidade do receptor em identificar quando uma determinada propriedade é satisfeita antes que o sistema atinja um estado em que ela não seja mais válida. A noção de TBP é comparada com a propriedade de utilidade e, é demonstrado que a noção proposta pode ser analisada sob diversos domínios e aplicações. Um método de verificação da TBP para indicar se um determinado sistema possui um conjunto de estados que satisfaz uma propriedade de interesse é proposto. Além disso, um estudo de caso para verificar a noção de TBP, no contexto de diagnóstico de falhas, e a *"current-state opacity"* é apresentado.

REFERÊNCIAS

Alguliyev, R., Imamverdiyev, Y., and Sukhostat, L. (2018). Cyber-physical systems and their security issues. *Computers in Industry*, 100, 212–223.

Barcelos, R.J. and Basilio, J.C. (2023). Ensuring utility while enforcing current-state opacity. In *IFAC World Congress*, volume 56, 4595–4600.

Cardoso, J.M.C., V. Moreira, M., and K. Carvalho, L. (2023). Synthesis of an obfuscation policy that guarantees utility satisfying a new privacy criterion. In *IFAC-PapersOnLine*. Elsevier.

Cassandras, C.G. and Lafortune, S. (2008). *Introduction to Discrete Event System*. Springer-Verlag New York, Inc., Secaucus, NJ.

Ding, D., Han, Q.L., Xiang, Y., Ge, X., and Zhang, X.M. (2018). A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 275, 1674–1683.

Dwork, C. (2006). Differential privacy. In *International Conference on Automata, Languages and Programming*.

Fritz, R., Fauser, M., and Zhang, P. (2019). Controller encryption for discrete event systems. In *2019 American Control Conference (ACC)*, 5633–5638. IEEE, Philadelphia, CA, USA.

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.

Lafortune, S., Lin, F., and Hadjicostis, C.N. (2018). On the history of diagnosability and opacity in discrete event systems. *Annual Reviews in Control*, 45, 257–266.

Li, X., Hadjicostis, C.N., and Li, Z. (2023). Opacity enforcement in discrete event systems using extended insertion functions under inserted language constraints. *IEEE Transactions on Automatic Control*.

Lima, P.M., da Silva, C.K., de Farias, C.M., Carvalho, L.K., and Moreira, M.V. (2022). Event-based cryptography for automation networks of cyber-physical systems using the stream cipher ChaCha20. *IFAC-PapersOnLine*, 55(28), 58–65. 16th IFAC Workshop on Discrete Event Systems WODES 2022.

Lin, F. (2011). Opacity of discrete event systems and its applications. *Automatica*, 47(3), 496–503.

Oliveira, S., B. Leal, A., Teixeira, M., and K. Lopes, Y. (2023). A classification of cybersecurity strategies in the context of discrete event systems. *Annual reviews in Control*.

Saboori, A. and Hadjicostis, C.N. (2007). Notions of security and opacity in discrete event systems. *Proceedings of the IEEE Conference on Decision and Control*, 5056–5061.

Saboori, A. and N. Hadjicostis, C. (2013). Verification of initial-state opacity in security applications of discrete event systems. *Information Sciences*, 246.

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

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1996). Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, 4(2), 105–124.

Simon, M.E. and Baldissera, F.L. (2023). Multi-robots coordination system for urban search and rescue assistance based on supervisory control theory. *Journal of Control Automation Electrical Systems*, 34, 484–495.

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: Theory and Applications*, 32.

Winterberg, A., Lafortune, S., Blischke, M., and Necmiye, O. (2022). A dynamic obfuscation framework for security and utility. *13th International Conference on Cyber-Physical Systems*.

Wu, Y.C., Raman, V., Rawlings, B.C., Lafortune, S., and Seshia, S.A. (2018). Synthesis of obfuscation policies to ensure privacy and utility. *Journal of Automated Reasoning*, 60(1), 107–131.

Yiding, J., Yi-Chin, W., and Stéphane, L. (2018). Enforcement of opacity by public and private insertion functions. *Automatica*, 93, 369–378.