Segurança de sistemas ciberfísicos contra ataques a atuadores: um método baseado em criptografia ⋆

Samuel Oliveira [*,**] **André B. Leal** [**] **Marcelo Teixeira** [***] **Yuri K. Lopes** [****]

[*] *Departamento de Ciências Exatas e Tecnológicas Universidade Federal do Amapá, Macapá/AP*
[**] *Programa de Pós-Graduação em Engenharia Elétrica Universidade do Estado de Santa Catarina, Joinville/SC*
[***] *Programa de Pós-Graduação em Engenharia Elétrica Universidade do Tecnológica Federal do Paraná, Pato Branco/PR*
[****] *Programa de Pós-Graduação em Computação Aplicada Universidade do Estado de Santa Catarina, Joinville/SC*

(emails: samuel.oliveira@unifap.br, andre.leal@udesc.br, mtex@utfpr.edu.br, yuri.lopes@udesc.br)

Abstract: Resilience against cyberattacks is an important requirement of modern cyber-physical systems (CPS) considering the industry 4.0 evolution. In this work, we use the formalism of Discrete Event Systems (DES) to design an approach that aims to protect cyber-physical systems against actuator attacks using event-based symmetric cryptography. In this approach, controllable events that are properly selected are encrypted before being transmitted from the controller to the plant, leading intruders to mistakenly infer the supervisor's control actions. Thus, it is presumed that attacks on actuators do not have the expected impact on the plant, preventing unsafe states from being reached. The proposed approach is applied to a case study from the literature, where the identification of the attack is done without causing any damage while maintaining the integrity of the system.

Resumo: Resiliência contra ataques cibernéticos é uma demanda de sistemas ciberfísicos modernos dada a evolução da indústria 4.0. Neste trabalho, formalismos de Sistemas a Eventos Discretos (SEDs) são utilizados para a formulação de uma abordagem para segurança de sistemas ciberfísicos contra ataques a atuadores através do uso de criptografia simétrica baseada em eventos. Nesta proposta, eventos controláveis devidamente selecionados são criptografados antes da sua transmissão do controlador para a planta, levando intrusos a inferir equivocadamente as ações de controle do supervisor. Assim, espera-se que ataques a atuadores não tenham o efeito esperado na planta, de maneira que estados inseguros não sejam alcançados. A abordagem proposta foi aplicada a um estudo de caso encontrado na literatura, em que a identificação de ataques ocorre sem que danos sejam infligidos, mantendo a integridade do sistema.

Keywords: Discrete event systems; cyber-physical systems; cybersecurity; cyberattacks; cryptography

Palavras-chaves: Sistemas a eventos discretos; sistemas ciberfísicos; cibersegurança; ataques cibernéticos; criptografia

1. INTRODUÇÃO

Sistemas ciberfísicos, do Inglês *Cyber-physical Systems* (CPS), são sistemas que utilizam componentes físicos e computacionais para executar e automatizar diversas tarefas. Nesses sistemas, sensores são utilizados para coletar dados do ambiente físico, enquanto atuadores são utilizados para a execução dos processos físicos. A troca de informação entre os componentes do sistema é realizada através de redes de comunicação, que apesar de sua eficácia, podem apresentar desafios, como o possível atraso na comunicação em situações de tráfego intenso de dados, além de questões relacionadas à segurança cibernética, como confidencialidade (equivalentemente, privacidade) e integridade de informações. Muitos destes sistemas possuem um conjunto finito de estados e transitam entre eles após a ocorrência de eventos discretos, e por isso são caracterizados como Sistemas a Eventos Discretos (SEDs). Dentre os modelos utilizados para controle de SEDs, destacam-se Autômatos Determinísticos de Estados Finitos (ADEF) e Redes de Petri (Cassandras e Lafortune, 2021).

Uma vez que redes de comunicação potencialmente vulneráveis são utilizadas para interconectar os componentes de um CPS, a confidencialidade e integridade de dados pas-

⋆ O presente trabalho foi realizado com apoio parcial da Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina - FAPESC 2023TR000333.

sam a ser requisitos importantes desses sistemas. Segundo Rashidinejad et al. (2019), invasores inteligentes (que se beneficiam do conhecimento prévio sobre o sistema) geralmente assumem um perfil não arriscado, priorizando a condução de *ataques furtivos* cujo objetivo é manter-se oculto ao sistema até que danos sejam causados. Ademais, Basilio et al. (2021) destacam que ataques contra a integridade de CPS podem ser conduzidos de maneira específica em (i) sensores, os quais afetam os dados observados na planta e enviados ao controlador; (ii) atuadores, que afetam os comandos de controle do controlador para a planta; e (iii) sensores e atuadores, que são aplicados nos canais de entrada e saída dos controladores de forma conjunta. Com base no exposto, este trabalho considera CPS sob ataques furtivos a atuadores, em que o intruso prioriza manter-se indetectável até que, em momento oportuno, intercepta e altera os comandos de controle enviados para a planta de forma que a integridade do sistema seja violada.

Na literatura, diversas estratégias de segurança são utilizadas para o desenvolvimento de mecanismos de defesa e ataque em SEDs. Uma delas é o uso de criptografia em prol da confidencialidade do sistema, como abordado no trabalho de Fritz et al. (2019), em que foi proposto um método que realiza a cifragem de todos os sinais de entrada e saída do controlador. A geração da chave criptográfica é realizada através da escolha de números inteiros grandes, de maneira que a fatoração (quebra da chave) se torne inviável. Contudo, o processo de criptografia resulta na transmissão de uma grande quantidade de dados, levando a problemas como a sobrecarga nos canais de comunicação. Já no trabalho de Lima et al. (2023) é proposto um método em que os eventos obervados na planta são cifrados antes da sua transmissão a fim de impedir que intrusos sejam capazes de inferir se uma sequência de eventos de interesse ocorreu no comportamento secreto do sistema. Neste caso, o modelo de criptografia realiza substituições aos eventos observados na planta, e portanto, não implica no aumento do volume dos dados transmitidos. Em contraste, Fritz e Zhang (2023) consideram o contexto de integridade de SEDs e propõe uma abordagem para detecção de ataques a sensores e atuadores baseada em um método de criptografia chamado de matriz de permutação, em que determinados eventos sofrem permutas nos canais de entrada e saída do controlador a fim de garantir ao sistema a capacidade de identificar ataques. Embora o método proposto não implique no aumento do volume dos dados transmitidos e possibilite a identificação de ataques, a existência de estados inseguros (em que danos ocorrem) no sistema não é considerada pelos autores.

A *síntese de ataque*, por sua vez, é uma estratégia baseada na perspectiva do intruso e tem como objetivo prover uma formulação de ataque de sucesso à integridade de SEDs com base no conhecimento prévio que o intruso possui sobre o sistema (Lin et al., 2020). Outra estratégia é a de *controle tolerante a ataques*, que se assemelha ao problema do controle tolerante a falhas (Moreira e Leal, 2020), em que o sistema deve ser capaz de diagnosticar toda e qualquer ocorrência de ataque antes de atingir um estado inseguro, para que medidas protetivas possam ser aplicadas e danos sejam evitados (Carvalho et al., 2018). Vale destacar que muitos sistemas possuem modelos que violam este princípio, o que significa que a identificação de ataques pode acontecer sem que haja a possibilidade de impedir a entrada em estados inseguros. Com esta perspectiva, a seguinte questão é levantada: *como proteger um sistema contra ataques a atuadores quando não há possibilidade de impedir que estados inseguros sejam alcançados após a detecção de ataques?* Nesse sentido, Ma e Cai (2022) investigaram a *síntese de supervisores resilientes*, que busca garantir, por meio do controle supervisório, que a planta seja impedida de atingir estados inseguros, mesmo que eventos controláveis sejam atacados. Neste contexto, o comportamento do sistema é restringido de tal forma que o supervisor resiliente impede o acesso a qualquer estado da planta que possua uma transição associada a um evento controlável que atinja um estado inseguro, sendo, portanto, uma abordagem conservadora. Embora seja garantido que o sistema não atinja estados inseguros, esta abordagem é restritiva, impedindo que o sistema possa executar parte de seu comportamento válido.

Neste trabalho, é proposta uma abordagem para proteção contra ataques a atuadores em sistemas que não possuem a capacidade de impedir a entrada em estados inseguros após a detecção de ataques, e também não podem sofrer restrições em seu comportamento válido, ainda que para fins de segurança. Esta abordagem foi principalmente inspirada no trabalho de Lima et al. (2023) que considerou o contexto de confidencialidade de SEDs. No presente trabalho, contudo, considera-se o contexto de integridade de SEDs, em que os eventos desabilitados pelo supervisor são cifrados com o objetivo de impedir que ataques a atuadores obtenham sucesso, permitindo a identificação de ataques sem que estados inseguros sejam atingidos.

As demais seções deste trabalho são organizadas da seguinte forma: A Seção 2 apresenta os conceitos básicos sobre autômatos e linguagens aplicados ao controle supervisório de SEDs. Na Seção 3 é apresentada a identificação do problema considerado. A Seção 4 apresenta a abordagem proposta. Na Seção 5, a abordagem proposta é aplicada a um estudo de caso da literatura. Por fim, na Seção 6 são apresentadas as conclusões e trabalhos futuros.

2. PRELIMINARES

Neste trabalho, o formalismo de autômatos é considerado para modelagem de SEDs. Segundo Cassandras e Lafortune (2021), um ADEF pode ser representado por uma quintupla $G = (X, \Sigma, f, x_0, X_m)$ em que X denota o seu conjunto finito de estados; Σ representa o conjunto de eventos que definem o alfabeto; $f : X \times \Sigma \to X$ a função de transição, possivelmente parcial; x_0 é o estado inicial do autômato, e por fim, X_m é o conjunto de estados marcados ou finais, de forma que $X_m \subseteq X$. Sendo G_1 e G_2 dois autômatos determinísticos, a composição síncrona (que unifica dois ou mais autômatos em um único) entre G_1 e G_2 é representada por $G_1 \| G_2$ (Cassandras e Lafortune, 2021).

De acordo com a teoria de controle supervisório (TCS) de SEDs, o conjunto de eventos Σ é particionado como $\Sigma_c \dot{\cup} \Sigma_{uc}$ que denotam os conjuntos de eventos controláveis e não-controláveis, respectivamente. Eventos controláveis permitem a desabilitação por parte do controlador, enquanto eventos não-controláveis se referem aos eventos observados na planta, e portanto, não podem ser desa-

bilidados. Nesse contexto, autômatos são utilizados para representar e manipular linguagens formais sobre o conjunto de eventos Σ, incluindo a palavra vazia ε. O Fecho de Kleene é uma operação denotada por Σ^* e representa o conjunto de todas as sequências finitas de elementos de Σ. A função de transição $f : X \times \Sigma$ é estendida recursivamente para $f : X \times \Sigma^*$, de forma que $f(x, \varepsilon) := x$ e $f(x, s\sigma) := f[f(x, s), \sigma]$ para $s \in \Sigma^*$ e $\sigma \in \Sigma$.

Um SED pode ser modelado através de um autômato G que representa duas linguagens: linguagem gerada ($L(G)$) e linguagem marcada ($L_m(G)$). A linguagem gerada $L(G) := \{s \in \Sigma^* \mid f(x_0, s)$ é definida $\}$ representa todos os comportamentos fisicamente possíveis e a linguagem marcada $L_m(G) := \{s \in L(G) \mid f(x_0, s) \in X_m\}$ representa os comportamentos que são fisicamente possíveis e que levam a um estado marcado, representando o completamento de tarefas. Assim, $L_m(G) \subseteq L(G)$. A notação $L(G)/s = \{t \in \Sigma^* : st \in L(G)\}$ se refere à continuação da linguagem $L(G)$ após uma sequência s.

Considere que G representa um sistema (planta) em malha aberta, isto é, um sistema livre de ações de controle. No contexto da TCS, a planta é controlada por um supervisor, representado por S, que possui a capacidade de observar os eventos gerados na planta e, então, desabilitar eventos controláveis com o objetivo de restringir o comportamento da planta com base em especificações de controle predefinidas. Dessa maneira, planta e supervisor compõem um novo sistema em malha fechada denotado por S/G, cuja linguagem gerada $L(S/G)$ é um subconjunto de $L(G)$, e a linguagem marcada $L_m(S/G)$ é um subconjunto de $L_m(G)$, considerando as ações de desabilitação impostas por S em G.

3. IDENTIFICAÇÃO DO PROBLEMA

Neste trabalho, abordamos ataques a atuadores, nos quais intrusos são capazes de sobrepor comandos de desabilitação de eventos controláveis, no âmbito do controle supervisório, por eventos de habilitação. Desse modo, o sistema é forçado a executar um comportamento diferente do especificado, que pode resultar em danos nos seus dispositivos físicos (Rashidinejad et al., 2019).

Neste contexto, considera-se que a planta G possui um conjunto de estados inseguros $X_B \subset X$, em que danos podem ocorrer, e um conjunto de estados seguros $X_{NB} = X \backslash X_B$, em que "\" denota a subtração de conjuntos [1]. O supervisor é projetado de forma a desabilitar eventos controláveis que levem a planta a um estado inseguro $x \in X_B$. Contudo, ataques de habilitação a esses eventos controláveis podem fazer com que o sistema atinja estados inseguros, conforme ilustrado na Figura 1. Com base no exposto, Carvalho et al. (2018) apresentaram um método para detecção de ataques de habilitação de atuadores, além de uma propriedade chamada *Controlabilidade Segura na Habilitação de Atuadores*, do Inglês, *Actuator Enablement Safe Controlabillity*. Essa propriedade é atendida caso seja possível detectar qualquer ocorrência de ataques e, em seguida, desabilitar um evento controlável antes que a planta atinja um estado inseguro. Em muitos casos, a

[1] As notações NB e B se referem aos termos *NOT BAD* e *BAD* que representam estados seguros e inseguros, respectivamente.

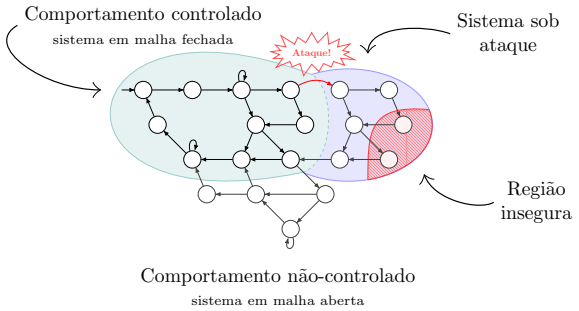
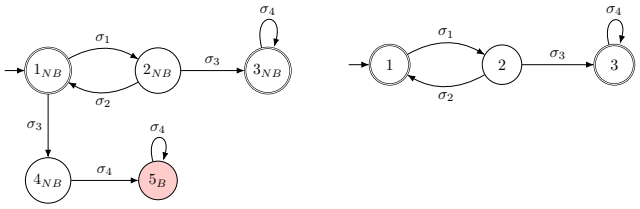


Figura 1. Sistema sob ação maliciosa de intrusos

depender dos modelos da planta, essa propriedade não é atendida, uma vez que a ocorrência de um ataque pode, de forma iminente, levar o sistema a atingir um estado inseguro, como apresentado no exemplo a seguir.

Exemplo 1. Considere o modelo de planta G apresentado na Figura 2a, em que $\Sigma_c = \{\sigma_1, \sigma_3\}$, e $\Sigma_{uc} = \{\sigma_2, \sigma_4\}$. Considere também o conjunto de estados seguros e inseguros $X_{NB} = \{1, 2, 3, 4\}$ e $X_B = \{5\}$, respectivamente. A função do supervisor (Fig. 2b) é desabilitar o evento controlável σ_3 no estado 1_{NB} a fim de impedir que a planta atinja o estado 5_B (estado inseguro).



(a) Planta G (b) Supervisor S

Figura 2. Modelos da planta e supervisor (Exemplo 1).

Uma vez que o intruso infere que o estado atual do sistema é o estado 1_{NB}, e identifica a emissão de um comando de desabilitação do evento σ_3, tem-se a certeza de que um ataque de habilitação a este evento conduzirá o sistema a atingir o estado inseguro 5_B. Além disso, o sincronismo entre planta e supervisor é comprometido, já que, após o ataque, a planta é levada a atingir o estado 4_{NB} e posteriormente o estado 5_B, enquanto o supervisor permanece no estado 1_{NB}.

3.1 Sistema sob ataque

Para analisar a segurança de SEDs sob ataques, supõe-se que o intruso tem conhecimento completo sobre os modelos da planta e do supervisor, e observa os eventos executados na planta. Além disso, assume-se, aqui, que o intruso tem a capacidade de identificar os eventos desabilitados pelo supervisor e pode sobrescrever comandos de desabilitação de eventos controláveis por comandos de habilitação. Já a estratégia de ataque considerada se baseia no modelo proposto por Lin et al. (2020), no qual o atacante se mantém encoberto e apenas realiza um ataque de habilitação quando está certo de que a planta atingirá um estado inseguro, caso contrário, o ataque seria diagnosticado pelo

sistema e o principal objetivo do intruso (manter-se encoberto até infligir danos) não seria alcançado. Com base no exposto, considera-se que os intrusos se limitam a atacar os eventos controláveis que rotulam transições que levam, iminentemente, a estados inseguros (eventos com potencial de danos).

Definição 1 (Conjunto de eventos com potencial de danos). Dado um SED representado por um autômato $G = (X, \Sigma, f, x_0, X_m)$, o conjunto de todos os eventos controláveis com potencial de danos é definido como $\Sigma_{c,d} := \{\sigma \in \Sigma_c : (\exists s \in L(S/G)$, para o qual $s\sigma \in L(G) \backslash L(S/G)) \wedge (\exists t \in L(G)/s\sigma : f(x_0, s\sigma t) \in X_B$, com $t \in \Sigma_{uc}^*)\}$.

Em detalhes, a Definição 1 estabelece que o conjunto de eventos com potencial de danos é composto por eventos controláveis existentes em estados seguros da planta e que, em algum dos caminhos após a sua ocorrência, existe uma cadeia formada apenas por eventos não-controláveis (ou pela palavra vazia ε) que leve a um estado inseguro.

O conjunto de todas as cadeias que levam o sistema a atingir estados inseguros é chamado de Linguagem de danos, como apresentado a seguir.

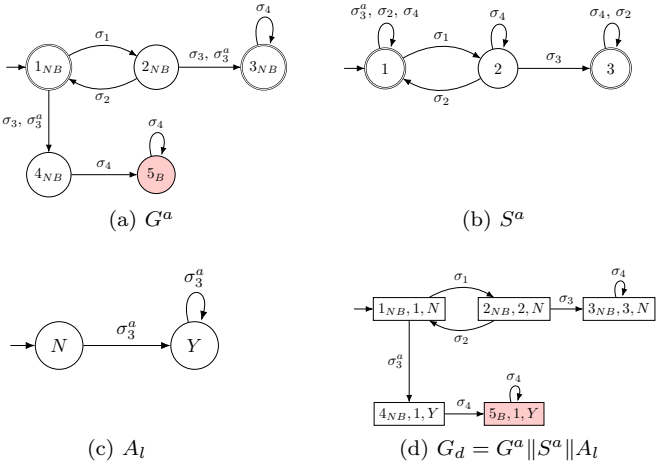
Definição 2 (Linguagem de danos). Seja $L(G)$ a linguagem gerada por um autômato G que representa um SED. A linguagem de danos é definida por $L_d(G) := \{s\sigma \in L(G) : (s \in L(S/G)$ e $\sigma \in \Sigma_{c,d}) \wedge (\exists t \in L(G)/s\sigma : f(x_0, s\sigma t) \in X_B$, com $t \in \Sigma_{uc}^*)\}$.

Desse modo, a linguagem de danos contém todas as cadeias de $L(G)$ em que o último evento controlável força o sistema a atingir estados inseguros. De acordo com o Exemplo 1, $L_d(G) = (\sigma_1\sigma_2)^*\sigma_3$.

Vale destacar que $L_d(G) \nsubseteq L(S/G)$, e portanto, nenhuma cadeia de $L_d(G)$ é executada no sistema em malha fechada, uma vez que o supervisor é projetado para evitar o alcance de estados inseguros. No Exemplo 1, o supervisor realiza desabilitações ao evento σ_3 a fim de impedir que o estado 5_B seja alcançado. Nesse sentido, o principal objetivo do intruso é forçar a execução de uma cadeia $s \in L_d(G)$ por meio de ataques de habilitação, para que estados inseguros sejam alcançados.

Seja $\Sigma_{c,d}^a := \{\sigma^a : \sigma \in \Sigma_{c,d}\}$ o conjunto de eventos com potencial de danos que são indevidamente habilitados pelo intruso a fim de alcançar um estado inseguro. Os eventos em $\Sigma_{c,d}^a$ são considerados não-controláveis, uma vez que representam as ações do atacante. Considera-se também que tais eventos são observáveis por parte do supervisor. Nesse sentido, para que o sistema identifique a ocorrência de ataques de habilitação, uma adaptação do método de detecção de ataques proposto por Carvalho et al. (2018), com base em diagnosticadores, foi utilizada. As etapas para obtenção do diagnosticador de ataques são apresentadas a seguir:

(1) Com base no modelo da planta G (Fig. 2a), construir um autômato G^a adicionando as possibilidades de ataques aos eventos em $\Sigma_{c,d}^a$. Especificamente, para cada transição rotulada por $\sigma \in \Sigma_{c,d}$, adiciona-se uma transição paralela σ^a que representa o ataque de habilitação, conforme ilustrado na Figura 3a;
(2) Construir um autômato S^a por intermédio da replicação do modelo do supervisor S (Fig. 2b), e adicionar *self-loops* para cada evento $\sigma^a \in \Sigma_{c,d}^a$ em todos os estados, exceto o estado cujo evento original σ faz parte de seu conjunto de eventos ativos. Semelhantemente, adicionar *self-loops* para cada evento $u \in \Sigma_{uc}$ em todos os estados em que o evento original não está disponível em seu conjunto de eventos ativos, a fim de modelar as possíveis ocorrências de eventos não-controláveis após ataques de habilitação, conforme ilustrado na Figura 3b;
(3) Construir um autômato rotulador A_l com dois estados rotulados por N e Y, respectivamente, além de transições rotuladas por eventos em $\Sigma_{c,d}^a$ entre o estado N e o estado Y, como apresentado na Figura 3c. Os rótulos dos estados servem para conferir ao diagnosticador a capacidade de indicar se ataques ocorreram (Y) ou não (N);
(4) Obter o modelo do diagnosticador de ataque G_d, realizando a composição síncrona dos modelos $G^a \| S^a \| A_l$, conforme mostrado na Figura 3d. Quando o diagnosticador alcança um estado cujo rótulo contém a notação Y, ele se torna certo de que um ataque ocorreu. Quando esse estado, ou um estado alcançado a partir dele com uma sequência de eventos não-controláveis apresentar o rótulo B, o diagnosticador indica que um estado inseguro foi alcançado.



(a) G^a (b) S^a (c) A_l (d) $G_d = G^a \| S^a \| A_l$

Figura 3. Modelos para obtenção do diagnosticador de ataques.

Ao analisar o diagnosticador obtido com base nos modelos apresentados no Exemplo 1, pode-se notar que, caso o intruso force o sistema a executar qualquer cadeia $s \in L_d(G)$ por meio de um ataque a σ_3^a, o ataque é diagnosticado, mas não há possibilidade de impedir que o estado inseguro seja alcançado, já que, após a deteção do ataque, não há um evento controlável que possa ser desabilitado antes que a planta atinja um estado inseguro, fato este que viola a propriedade de controlabilidade segura.

4. ABORDAGEM PROPOSTA

O presente trabalho visa prover uma abordagem para segurança contra ataques a atuadores em SEDs, cujos modelos de planta não mantêm a propriedade de controlabilidade segura, e que não podem sofrer restrições em seu comportamento controlado. Essas características são comuns em sistemas de manufatura, como no estudo de caso a ser

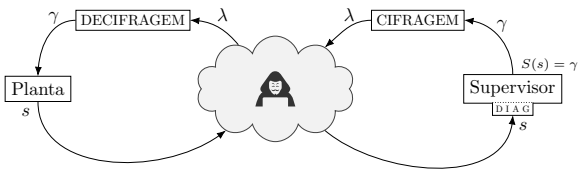


Figura 4. Arquitetura de um sistema ciberfísico com cifragem de eventos controláveis.

apresentado na Seção 5. Esta abordagem é baseada em um método de criptografia simétrica chamada de *cifra de substituição* (Shimeall e Spring, 2014), em que os eventos controláveis são cifrados antes da sua transmissão para a planta, que por sua vez, possui um módulo capaz de decifrar os eventos recebidos. Além disso, um diagnosticador de ataques é utilizado, permitindo que os ataques aos atuadores sejam reconhecidos pelo sistema. Com o uso desta abordagem, busca-se frustrar a ação maliciosa de intrusos, além de permitir a identificação de ataques sem que estados inseguros sejam atingidos. A arquitetura de um sistema ciberfísico com a cifragem de eventos controláveis é apresentada na Figura 4.

Vale destacar que o método de criptografia utilizado não exige a adoção de chaves formadas por inteiros grandes, e consequentemente, não aumenta a quantidade de dados enviados para a rede. Ao invés disso, os comandos de controle enviados pelo supervisor são substiuídos de forma que o intruso é levado a inferir erroneamente os eventos que estão sendo desabilitados. Dessa maneira, os ataques de habilitação aos eventos controláveis não terão o efeito esperado na planta.

4.1 Cifras de substituição

O método de criptografia baseado na cifra de substituição codifica uma cadeia (palavra) através da substituição de cada símbolo da cadeia por um símbolo diferente do alfabeto, conforme indicado por uma chave k, cujo valor representa o deslocamento nos símbolos do alfabeto (Shimeall e Spring, 2014). Para realizar a substituição, um índice i (número inteiro) é associado a cada símbolo do alfabeto, denotado por Σ. A função de cifragem f_c de um símbolo $\sigma_i \in \Sigma$ pode ser descrita matematicamente como $f_c(\sigma_i) = \sigma_{(i+k)\%n}$, tal que "%" denota a operação módulo, que encontra o resto da divisão de "$(i + k)$" por "n", que, por sua vez, representa a cardinalidade de Σ. Semelhantemente, a função de decifragem pode ser descrita matematicamente como $f_d(\sigma_i) = \sigma_{(i-k)\%n}$. A Figura 5 ilustra o uso da cifra de substituição nos símbolos do alfabeto $\Sigma = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ com $k = 4$.

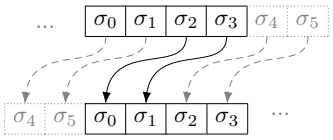


Figura 5. Função de cifragem $f_c(\sigma_i)$ com $k = 4$.

4.2 Cifragem de eventos controláveis em SEDs sob ataques

A abordagem proposta aplica este modelo de criptografia em um sistema em malha fechada S/G, com base no conjunto de eventos controláveis da planta G, para que o intruso seja levado a inferir, equivocadamente, os eventos desabilitados pelo supervisor. Para tanto, a função de cifragem de símbolos f_c é aplicada em cadeias por meio de um operador de projeção criptográfica $P_c : L(G) \to \Sigma^*$ que projeta uma cadeia $s \in L(G)$ em uma cadeia cifrada $s' \in \Sigma^*$, e é definido, recursivamente, como $P_c(s\sigma) = P_c(s)P_c(\sigma)$, para todo $s \in L(G)$ e $\sigma \in \Sigma$, com as seguintes propriedades:

$$P_c(\varepsilon) = \varepsilon,\ P_c(\sigma) = \begin{cases} f_c(\sigma), & \text{se } \sigma \in \Sigma_c \\ \sigma, & \text{se } \sigma \in \Sigma_{uc} \end{cases} \quad (1)$$

Para a obtenção de uma linguagem cifrada, a projeção criptográfica P_c é estendida para linguagens, de maneira que todas as sequências de $L(G)$ são submetidas à projeção P_c. Assim, a operação $P_c(L(G))$ resulta em uma linguagem cifrada, denotada por $L_c(G)$, e é definida como:

$$P_c(L(G)) = \{t \in \Sigma^* : (\exists s \in L(G))[\, P_c(s) = t]\,\} \quad (2)$$

Dessa maneira, a projeção criptográfica da linguagem $L(G)$ consiste em substituir todos os eventos controláveis $\sigma \in \Sigma_c$ contidos em todas as suas cadeias, por eventos diferentes.

Ao considerar um sistema em malha fechada S/G, os eventos desabilitados pelo supervisor são submetidos à projeção criptográfica. Contudo, exige-se que, na planta, a substituição seja revertida para que as ações de controle do supervisor tenham o efeito esperado. A reversão de uma cadeia cifrada pode ser realizada através do operador de projeção criptográfica inversa $P_c^{-1} : \Sigma^* \to L(G)$, que é definido recursivamente como $P_c^{-1}(s\sigma) = P_c^{-1}(s)P_c^{-1}(\sigma)$, para todo $s \in L_c(G)$ e $\sigma \in \Sigma$ e possui as seguintes propriedades:

$$P_c^{-1}(\varepsilon) = \varepsilon,\ P_c^{-1}(\sigma) = \begin{cases} f_d(\sigma), & \text{se } \sigma \in \Sigma_c \\ \sigma, & \text{se } \sigma \in \Sigma_{uc} \end{cases} \quad (3)$$

Considere o uso da projeção criptográfica na linguagem gerada pelo modelo de planta G apresentado no Exemplo 1 (Fig. 2a), em que $\Sigma_c = \{\sigma_1, \sigma_3\}$ representa o alfabeto a ser cifrado com uma chave $k = 1$. Logo, σ_1 será cifrado como σ_3, e vice-versa. O autômato que modela a linguagem cifrada $L_c(G)$ é apresentado na Figura 6.

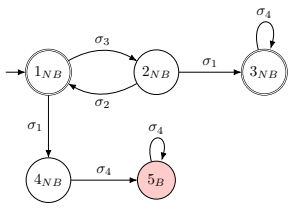


Figura 6. Autômato que modela a linguagem cifrada $L_c(G)$.

Note que, quando a sequência $\sigma_1\sigma_2$ ocorre, o intruso observa a sequência transmitida $\sigma_3\sigma_2$. Logo, o intruso não é capaz de estimar o estado atual do sistema, já que não há cadeias em $L(G)$ que contenham a subcadeia $\sigma_3\sigma_2$. É importante destacar que, ao se deparar com a desconformidade entre a sequência de eventos observada e as sequências possíveis em $L(G)$, o intruso pode ser levado a questionar seu conhecimento sobre os modelos da planta, bem como a supor a existência de algum método

de segurança que justifique tal observação. Ainda assim, o intruso não será capaz de realizar um ataque com a certeza de que o sistema atingirá um estado inseguro.

Em aplicações práticas, uma possível forma de reduzir esta desconformidade é reduzir o conjunto de eventos controláveis utilizados no processo de cifragem, sendo necessário avaliar se o intruso será mantido incapaz de inferir se ataques levarão o sistema a atingir estados inseguros. Isto pode ser alcançado ao explorar a modularidade do sistema, quando possível, utilizando abordagens formais para a síntese de supervisores modulares, como a abordagem modular local (Queiroz e Cury, 2000). Neste contexto, supervisores locais atuam em conjunto sobre plantas locais por meio de uma lógica de controle minimamente restritiva e não bloqueante. Dessa maneira, em vez de aplicar a cifragem baseada no conjunto de eventos controláveis do sistema global, o conjunto de eventos a ser criptografado se limita aos modelos de planta local que possuem estados inseguros, reduzindo, assim, a desconformidade entre a sequência de eventos observada pelo intruso e as sequências possíveis de serem executadas no sistema.

É importante destacar que neste trabalho, assim como no trabalho de Lima et al. (2023), não são considerados métodos de criptoanálise, considerando a limitação de espaço, bem como o fato de que a aplicação do método proposto atinge o objetivo de identificar ataques sem que estados inseguros sejam atingidos, como apresentado no estudo de caso a seguir.

5. ESTUDO DE CASO

O exemplo de aplicação prática considerado no presente trabalho é inspirado no sistema de mistura de fluídos apresentado por Leitão et al. (2020) para fins de diagnose de falhas. O sistema é composto por três válvulas, uma bomba e um sensor de concentração, como ilustrado na Figura 7a. Seu funcionamento ocorre da seguinte maneira: inicialmente a Válvula A é aberta por 9 segundos para encher o tanque com um solvente. Na sequência, a Válvula A é fechada e a Válvula C abre para encher o tanque com soluto. Quando a mistura atinge o nível de concentração desejado, o sensor de concentração emite um sinal para que o processo seja finalizado com o fechamento da Válvula C. Em seguida, a drenagem do tanque é realizada abrindo a Válvula D e acionando a bomba, que é desligada apenas quando o sensor indicar que o tanque está vazio, permitindo um novo clico de operação.

Os modelos dos elementos da planta do sistema de mistura de fluídos são ilustrados na Figura 7b e as descrições dos eventos usados na modelagem são apresentadas na Tabela 1. O supervisor monolítico é ilustrado na Figura 7c, conforme apresentado por Leitão et al. (2020).

5.1 Modelagem do sistema sob ataque

Por considerar a manutenção da integridade do sistema em face de ataques a atuadores, o foco deste trabalho está na etapa de drenagem do fluído, uma vez que a bomba só deve ser acionada após a abertura da Válvula D, caso contrário, danos podem ocorrer. Do mesmo modo, enquanto a bomba estiver ligada, a Válvula D deverá permanecer aberta. Dessa forma, se um intruso for capaz

Tabela 1. Eventos da planta.

Evento	C/NC	Descrição
VA_{open}	C	Comando para abrir a válvula A
VA_{close}	C	Comando para fechar a válvula A
T_{9sec}	NC	Sinal do *timer* de 9 segundos
VC_{open}	C	Comando para abrir a válvula C
VC_{close}	C	Comando para fechar a válvula C
INF_{sup}	NC	Sinal do sensor de concentração
VD_{open}	C	Comando para abrir a válvula D
VD_{close}	C	Comando para fechar a válvula D
P_{start}	C	Comando para ligar a bomba
P_{stop}	C	Comando para desligar a bomba
SUP_{inf}	NC	Sinal que indica que o tanque está vazio

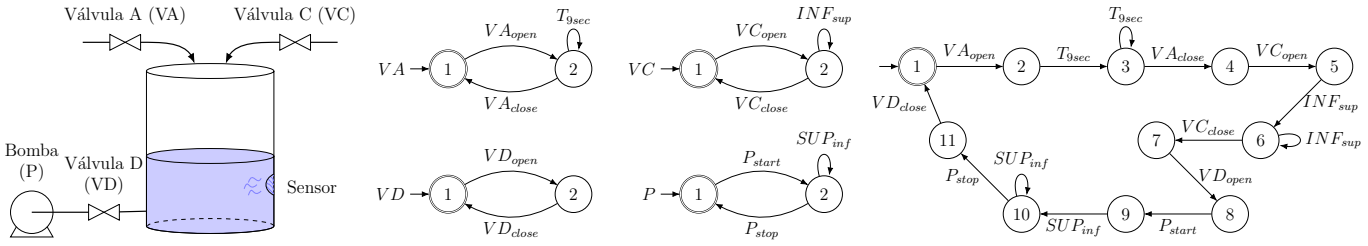
de provocar o fechamento da Válvula D enquanto a bomba estiver em funcionamento, bem como ativar a bomba enquanto a Válvula D estiver fechada, o sistema atingirá um estado inseguro, podendo ocasionar danos na bomba.

A modelagem do sistema sob ataque baseou-se na síntese de supervisores modulares locais, em que subsistemas da planta são agrupados em plantas locais de acordo com especificações de controle predefinidas. Por conta da limitação do espaço neste artigo, os modelos das especificações de controle foram omitidos. No entando, destaca-se que as especificações relacionadas ao controle das válvulas VA, VC e VD (G_1, G_2 e G_3) e da bomba (G_4), respectivamente, foram utilizadas para a obtenção de dois supervisores modulares locais S^1_{loc} e S^2_{loc}, os quais são ilustrados na Figura 8. Vale destacar que, como esperado, a composição síncrona entre os modelos $S^1_{loc} \parallel S^2_{loc}$ resulta em um autômato indêntico ao supervisor monolítico ilustrado na Figura 7c. O modelo da planta $G^1_{loc} = G_1 \| G_2 \| G_3$, com 8 estados, foi omitido pela limitação de espaço. Já o modelo da planta $G^2_{loc} = G_3 \| G_4$ é apresentado na Figura 8c com os conjuntos de estados inseguros e seguros $X_B = \{2\}$ e $X_{NB} = \{1, 3, 4\}$, respectivamente.

Do ponto de vista do intruso, a planta G^2_{loc} poderia ser forçada a antingir o estado inseguro 2_B através da execução de qualquer cadeia da linguagem de danos $L_d(G^2_{loc})$, o que pode ser alcançado por meio de ataques de habilitação ao evento P_{start} no estado 1_{NB}, e ao evento VD_{close} no estado 4_{NB}. Assim, os eventos P_{start} e VD_{close} compõem o conjunto de eventos com potencial de danos, isto é, $\Sigma_{c,d} = \{P_{start}, VD_{close}\}$.

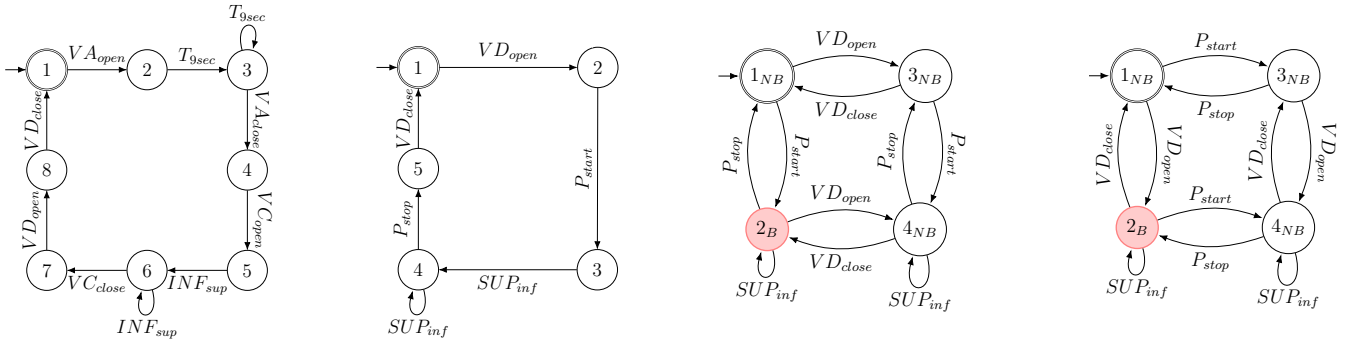
Para evitar a execução de qualquer cadeia $s \in L_d(G^2_{loc})$, o conjunto de eventos controláveis da planta local G^2_{loc} passa a ser utilizado como base para a função de cifragem f_c, de forma que os eventos em $\Sigma_{c,d}$ são substituídos pelos demais eventos $\sigma \in \Sigma_c \backslash \Sigma_{c,d}$ para que as ações de intrusos não causem o efeito esperado na planta. Ademais, as linguagens de ambas as plantas locais $L(G^1_{loc})$ e $L(G^2_{loc})$ são submetidas à projeção criptográfica, visto que englobam o modelo do subsistema da Válvula D (G_3), e possuem o evento $VD_{close} \in \Sigma_{c,d}$. Diante disso, considere o conjunto de eventos controláveis de $G^2_{loc} = \{VD_{open}, VD_{close}, P_{start}, P_{stop}\}$ e uma chave $k = 2$. O autômato que modela a linguagem cifrada $L_c(G^2_{loc})$ é apresentado na Figura 8d.

Dado que ataques aos eventos controláveis com potencial de danos são decifrados na planta como eventos diferentes, é necessário diagnosticar tanto ataques aos eventos em $\Sigma_{c,d}$, como ataques aos eventos que os substituem. Dessa maneira, todos os supervisores modulares locais que



(a) Ilustração do sistema. (b) Modelos dos subsistemas da planta. (c) Supervisor monolítico.

Figura 7. Sistema de mistura de fluídos. Adaptado de Leitão et al. (2020).



(a) S^1_{loc} (b) S^2_{loc} (c) G^2_{loc} (d) $L_c(G^2_{loc})$

Figura 8. Modelos dos supervisores modulares locais (a) e (b), planta local G^2_{loc} com estado inseguro 2_B (c) e linguagem cifrada $L_c(G^2_{loc})$ (d).

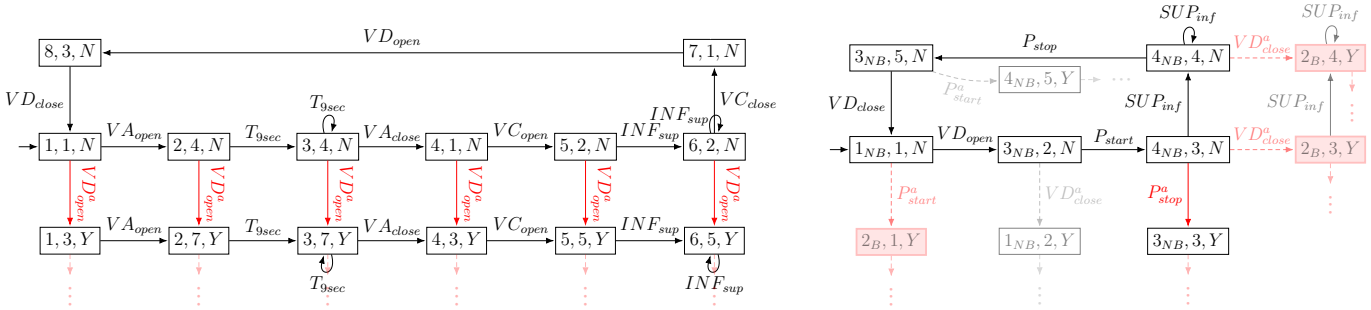
lidam com eventos controláveis em $\Sigma_{c,d}$ devem possuir um respectivo diagnosticador de ataques. Assim, S^1_{loc} e S^2_{loc} agregam os diagnosticadores G^1_d e G^2_d, respectivamente. Tais diagnosticadores são obtidos através das etapas apresentadas na Seção 3, em que $G^1_d = G^{1,a}_{loc} \parallel S^{1,a}_{loc} \parallel A_l$, e $G^2_d = G^{2,a}_{loc} \parallel S^{2,a}_{loc} \parallel A_l$. Os diagnosticadores de ataques são apresentados na Figura 9. Os eventos atacados pelo intruso são representados por transições com linhas pontilhadas. Já os eventos decifrados na planta em decorrência de ataques são representados pelas transições com linhas contínuas em vermelho.

Observe que nos seis primeiros estados de G^1_d e no primeiro estado de G^2_d, os eventos VD_{open} e P_{start} são, respectivamente, desabilitados pelos supervisores S^1_{loc} e S^2_{loc}. Na perspectiva do intruso, ataques ao evento P_{start} nos estados mencionados levariam o sistema a atingir o estado inseguro 2_B de forma imediata. Contudo, ao considerar o uso da cifragem de eventos controláveis, observa-se que os ataques ao evento P_{start} são decifrados na planta como uma habilitação ao evento VD_{open}, o que faz com que a planta execute um comportamento que viola as especificações de controle, mas evita que o estado inseguro 2_B seja alcançado. Além disso, a habilitação do evento VD_{open} nestes estados leva o diagnosticador G^1_d a atingir um estado certo de que o ataque ocorreu.

Observe também que nos estados iniciais de G^1_d, a sequência de eventos observada pelo intruso não possui qualquer desconformidade com as sequências possíveis de ocorrer em $L(S^2_{loc}/G^2_{loc})$, já que ambos eventos P_{start} e VD_{open} são desabilitados, e que o evento P_{start} é decifrado como VD_{open}, e vice-versa. Ou seja, o intruso não é capaz de determinar a existência de qualquer mecanismo de proteção que realize substuições em eventos controláveis. Por outro lado, quando a etapa de drenagem é iniciada e a sequência $VD_{open}\,P_{start}$ é executada no sistema, o intruso observa a cadeia cifrada $P_{start}\,VD_{open}$, identificando a desconformidade com a linguagem do sistema em malha fechada, que não contém a sequência observada. Contudo, ainda que o intruso pressupunha que o estado 4_{NB} é o estado alcançado em G^2_{loc} após a ocorrência da cadeia $P_{start}\,VD_{open}$, o ataque de habilitação ao evento VD_{close}, que levaria o sistema a atingir o estado inseguro 2_B, será interpretado na planta como habilitação ao evento P_{stop}, fato este que implica na violação da especificação de controle, mas torna a ação maliciosa infrutífera, impedindo que o estado 2_B seja alcançado.

Ao analisar o diagnosticador G^2_d, observa-se que ataques ao evento VD_{close} no estado $4_{NB}, 3, N$ são diagnosticados, pois o evento P_{stop} é habilitado na planta após a decifragem do ataque, levando o diagnosticador a atingir o estado $3_{NB}, 3, Y$. Contudo, após a ocorrência do evento não-controlável SUP_{inf}, o diagnosticador atinge o estado $4_{NB}, 4, N$, em que o ataque deixa de ser diagnosticado, pois o evento P_{start} habilitado na planta em decorrência do ataque é o mesmo evento habilitado pelo supervisor. De qualquer maneira, a abordagem proposta impede o alcance de estados inseguros em decorrência do ataque.

Destaca-se que o diagnosticador G^2_d é capaz identificar ataques que não seguem a estratégia de Lin et al. (2020), e portanto, não resultam em danos. Tais ataques são representados na Figura 9 por transições com linhas pontilhadas em cinza nos estados $3_{NB}, 2, N$ e $3_{NB}, 5, N$.



(a) Parte do diagnosticador G_d^1

(b) Parte do diagnosticador G_d^2

Figura 9. Diagnosticadores de ataques.

É importante enfatizar que, mesmo com o uso da abordagem proposta, os ataques de habilitação a eventos controláveis imputam algum nível de disrupção na planta, violando especificações de controle e afetando o sincronismo entre planta e supervisor. Este problema pode ser mitigado através de medidas de recuperação, como o método proposto por Alves e Pena (2022), em que a recuperação do sistema é alcançada por meio do uso de autômatos sincronizáveis.

6. CONCLUSÃO

Neste trabalho foi considerado o problema da segurança de sistemas ciberfísicos modelados como SEDs em que intrusos são capazes de realizar ataques de habilitação de eventos controláveis a fim de que estados inseguros sejam atingidos na planta, e danos sejam provocados. A abordagem proposta é baseada em criptografia simétrica de eventos controláveis (cifras de substuição) em que os ataques realizados por intrusos são submetidos à uma função de decifragem antes de sua execução na planta, causando um efeito diferente do esperado pelo intruso e permitindo ao sistema identificar a ocorrência de ataques sem que estados inseguros sejam atingidos.

A abordagem proposta foi aplicada em um estudo de caso disponível na literatura, sendo um sistema de mistura de fluídos. A modelagem do sistema baseou-se na síntese de supervisores modulares locais, de forma que apenas os eventos controláveis da planta local que contém estados inseguros são utilizados no processo de criptografia.

Como perspectivas de trabalhos futuros, destaca-se o desenvolvimento de um método para a seleção ótima de um conjunto mínimo de eventos para o processo de criptografia de forma que a segurança contra ataques de habilitação seja mantida e a desconformidade entre as sequências de eventos observadas pelo intruso e as sequências contidas na linguagem do sistema em malha fechada seja reduzida. Pretende-se ainda considerar estudos de caso em que transições com eventos não-controláveis atinjam estados inseguros e ataques a sensores e atuadores são realizados.

REFERÊNCIAS

Alves, L.V.R. e Pena, P.N. (2022). Secure recovery procedure for manufacturing systems using synchronizing automata and supervisory control theory. *IEEE Transactions on Automation Science and Engineering*, 19(1), 486–496.

Basilio, J.C., Hadjicostis, C.N., e Su, R. (2021). Analysis and control for resilience of discrete event systems: Fault diagnosis, opacity and cyber security. *Foundations and Trends in Systems and Control*, 8(4), 285–443.

Carvalho, L.K., Wu, Y.C., Kwong, R., e Lafortune, S. (2018). Detection and mitigation of classes of attacks in supervisory control systems. *Automatica*, 97, 121–133.

Cassandras, C.G. e Lafortune, S. (2021). *Introduction to Discrete Event Systems*. Springer Cham, 3 edition.

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

Fritz, R. e Zhang, P. (2023). Detection and localization of stealthy cyber attacks in cyber-physical discrete event systems. *IEEE Transactions on Automatic Control*, 1–8.

Leitão, H.A.S., Rosso, R.S.U., Leal, A.B., e Zoitl, A. (2020). Fault handling in discrete event systems applied to iec 61499. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, 1039–1042.

Lima, P.M., Carvalho, L.K., e Moreira, M.V. (2023). Ensuring confidentiality of cyber-physical systems using event-based cryptography. *Information Sciences*, 621, 119–135.

Lin, L., Zhu, Y., e Su, R. (2020). Synthesis of covert actuator attackers for free. *Discrete Event Dynamic Systems: Theory and Applications*, 30(4), 561 – 577.

Ma, Z. e Cai, K. (2022). On resilient supervisory control against indefinite actuator attacks in discrete-event systems. *IEEE Control Systems Letters*, 6, 2942–2947.

Moreira, B.G. e Leal, A.B. (2020). A proposal for an active diagnoser for safe fault-tolerant control of discrete event systems. *IFAC-PapersOnLine*, 53(4), 282–287.

Queiroz, M.H.D. e Cury, J.E. (2000). Modular supervisory control of large scale discrete event systems. In *Discrete event systems*, 103–110. Springer.

Rashidinejad, A., Wetzels, B., Reniers, M., Lin, L., Zhu, Y., e Su, R. (2019). Supervisory control of discrete-event systems under attacks: An overview and outlook. In *2019 18th European Control Conference (ECC)*, 1732–1739.

Shimeall, T.J. e Spring, J.M. (2014). Chapter 8 - resistance strategies: Symmetric encryption. In T.J. Shimeall e J.M. Spring (eds.), *Introduction to Information Security*, 155–186. Syngress, Boston.