Secure Goal-Oriented Communication: Defending Against Eavesdropping Timing Attacks

Federico Mason, *Member, IEEE*, Federico Chiariotti, *Senior Member, IEEE*, Pietro Talli, *Graduate Student Member, IEEE*, and Andrea Zanella, *Senior Member, IEEE*

***Abstract*—Goal-oriented Communication (GoC) is a new paradigm that activates data transmission only when it is instrumental for the receiver to achieve a certain goal. This leads to the advantage of reducing the frequency of transmissions significantly while maintaining adherence to the receiver's objectives. However, GoC scheduling also opens a timing-based side channel that an eavesdropper can exploit to estimate the state of the system. This type of attack sidesteps even information-theoretic security, as it exploits the timing of updates rather than their content. In this work, we study such an eavesdropping attack against pull-based goal-oriented scheduling for remote monitoring and control of Markov processes. We provide a theoretical framework for defining the effectiveness of the attack and propose possible countermeasures, including three heuristics that provide a balance between the performance gains offered by GoC and the amount of leaked information. Our results show that, while a naive GoC scheduler allows the eavesdropper to correctly guess the system state about 60% of the time, our heuristic defenses can halve the leakage with a marginal reduction of the benefits of goal-oriented approaches.**

***Index Terms*—Goal-oriented communication, eavesdropping, timing attacks, hidden Markov models.**

I. Introduction

OVER the past few years, the GoC paradigm has attracted a significant amount of interest from the research community. The concept was advanced by Shannon and Weaver in his 1949 introduction to Shannon's theory of communication [1], and regards the design of more advanced communication protocols that go beyond the mere transmission of bits and consider the meaning and usefulness of the data for the receiver in the decision over what and when to transmit. However, practical implementation of these ideas requires powerful machine learning techniques [2] and, therefore, has only recently become feasible.

Goal-oriented approaches were initially applied to compression [3] and have been subsequently extended to scheduling strategies that consider contextual and past information [4]. These initial studies have shown that GoC leads to impressive performance advantages and may become a key technology in 6G [5], as it can exploit structural information in the data, relevance metrics for the task, and the statistics of the communication channel to significantly reduce resource usage. However, the privacy of GoC suffers the same fragility observed in semantic communication [6], as binding communication decisions to the relevance of the data inherently exposes an attack surface for eavesdropping attacks [7]. The most common approach to enhance GoC or semantic communication security is to train the transmitter to encrypt the data [8], modifying the encoding mechanism to trigger an incorrect semantic interpretation by possible eavesdroppers [9], while allowing the intended receiver to decode the original message. Information-theoretic approaches [10] can provide more solid confidentiality guarantees [11], but only under specific assumptions on the nature of the encoder and decoder.

The use of GoC introduces another aspect: while semantic communication schemes often focus on individual packets, monitoring or controlling a Cyber-Physical System (CPS) has a strong timing component [12], and deciding *when* to transmit an update and how to schedule resources [13] is as important as deciding *what* to transmit [14]. This is particularly relevant in Internet of Things (IoT) scenarios or other resource-constrained monitoring systems, where GoC can be used to reduce the frequency of updates according to the status of the monitored process.

This timing component has so far been considered exclusively as an advantage [14], as it reduces the amount of transmitted data and, consequently, energy consumption and traffic. However, systems that adapt the scheduling of updates to the conditions of the monitored process open a side channel that can be exploited by an attacker aiming to infer the system state from the *timing* of messages [15]. In these scenarios, timing attacks can leak information about the state of the process even when the content of transmitted packets is information-theoretically secure, thus eluding content-based semantic security mechanisms and countermeasures.

In this work, we address timing attacks against GoC that, to the best of our knowledge, have not yet been considered in the literature. We focus on a state-of-the-art pull-based GoC scheme, in which a controller node maintains an online estimate of the state of a remote Markov process, requesting updates from a remote sensor that can observe the process directly [14]. We then model an eavesdropping attack that

Received 13 July 2025; revised 22 December 2025; accepted 24 February 2026. Date of publication 3 March 2026; date of current version 10 March 2026. This work was supported in part by the National Recovery and Resilience Plan (NRRP) funded by the European Union NextGenerationEU Project as part of the "Research and Innovation on Futures Telecommunications Systems and Networks, to make Italy more Smart (RESTART)" Partnership under Grant PE0000001; and in part by the High-Performance Computing Infrastructure Developed under the Project "Comunità Veneta Per Il Calcolo Scientifico (CONVECS)," funded by the Programma Regionale (PR) Veneto Fondo Europeo di Sviluppo Regionale (FESR) 2021–2027 Program, Priority 1–Specific Objective 1.1–Action 1.1.2. *(Corresponding author: Federico Mason.)*

The authors are with the Department of Information Engineering, University of Padua, 35131 Padua, Italy (e-mail: federico.mason@unipd.it; federico.chiariotti@unipd.it; pietro.talli@phd.unipd.it; andrea.zanella@unipd.it).

Digital Object Identifier 10.1109/JSAC.2026.3669875

exploits the timing side channel, analyzing its capability to infer the actual state of the remote process under the assumption of perfect secrecy of the message content. In fact, even if individual messages are undecipherable by the attacker, the timing between communications provides a rich source of indirect information, leading to a significant leakage.

We then consider possible defenses that the legitimate nodes may employ when adopting a *differential privacy* approach [16], i.e., easing the absolute secrecy requirement but strictly bounding the mutual information that an attacker can exploit. We can then control the trade-off between secrecy and performance: while the benefits of GoC are inextricably tied to the side channel that it opens, we can control the trade-off between secrecy and performance, allowing system designers to select an operating point by adopting a proactive defense strategy. We present three such strategies in this work, analyzing their ability to manage this trade-off.

To our knowledge, this manuscript is the first to consider the secrecy implications of timing attacks against GoC, and includes the following main contributions:

- We provide a rigorous model of timing attacks in GoC, defining information leakage as a function of the time for which confidentiality must be ensured.
- We prove that finding a game-theoretical equilibrium when both the legitimate agent and the eavesdropper are rational actors is a computationally hard problem, whose complexity grows exponentially with the state space size of the control problem.
- We propose three heuristic defense schemes that can effectively control the trade-off between secrecy and performance in the considered scenarios, analyzing their performance and computational complexity.
- We evaluate the effectiveness of timing attacks and defensive strategies through Monte Carlo simulations for both estimation and control scenarios, offering a preliminary characterization of the empirical Pareto frontier that maps the design space for secure control.
- We present a hybrid attack that combines imperfect message secrecy and timing side channel, showing how the eavesdropper can take advantage of the timing attack to significantly increase the information leakage.

A preliminary version of this work was presented as a conference paper in [17]. This manuscript extends our previous results in many directions, including the definition and analysis of the new Packing Defense from Eavesdropping (PDE) and Stochastic Defense from Eavesdropping (SDE) policies, extending the model to stochastic strategies and the analysis of the framework in the case of control applications. The analysis of the combined attack on both message content and transmission timing, presented in the Appendix, is another novel contribution of the present work. The latter is tangential to our primary focus on timing side channels, offering a characterization of a more complex attack surface.

The remainder of the paper is organized as follows. First, Sec. II reviews state-of-the-art security schemes in semantic and GoC communication. Hence, Sec. III presents the GoC model, drawing from the results of our previous work [14], while Sec. IV presents the eavesdropping attack and the game-theoretical framework. Subsequently, Sec. V introduces the heuristic algorithms to mitigate information leakage in the system, and Sec. VI discusses our simulation settings and results. Finally, Sec. VII concludes the article and describes possible avenues for future research.

II. Related Work

As GoC is still a relatively new paradigm, research on its security aspects, such as eavesdropping attacks, is still in its infancy. The existing GoC security literature mostly focuses on a subclass of GoC problems that focus on reconstructing the transmitted information directly, without any memory or time-dependence. In this context, timing attacks are not meaningful, and the focus is on the content of each message.

In addition to early work using an information bottleneck approach [10], previous studies mainly deal with eavesdropping attacks using deep learning [7]. More recently, the work in [11] has provided a near-information-theoretic security approach for semantic communication. The authors consider a legitimate receiver with a higher Signal to Noise Ratio (SNR) than the eavesdropper, allowing the semantic scheme to exploit this advantage by properly encoding the semantic symbols.

A very common semantic communication approach is deep Joint Source-Channel Coding (JSCC). This model was adapted to include Shannon secrecy in [8], extending the information-theoretic approach to learning-based semantic encoders, whose constellations are learned rather than hand-designed: in this case, the learning algorithm converges to a secret semantic encoding by using secrecy as an additional objective, exploiting similar principles as traditional information-theoretic security. Interestingly, the JSCC protection module can be implemented before [18] or after semantic encoding [8], or integrated within the encoder [19], [20], with similar results and trade-offs in terms of secrecy and image transfer quality.

Another example of semantic encryption is given in [9], where eavesdroppers adopt a model inversion approach to retrieve the original information. The use of explicit semantic features of the image [21] can also be used to generate shared secrets between the transmitter and the legitimate receiver that can be used to improve security. The same concept has been extended to the vision transformer architecture in [22]. Finally, the authors of [23] adopt steganographic techniques to fool the eavesdropper into recovering an unrelated image, while keeping the meaningful content secure.

Active attacks that go beyond eavesdropping have been designed and tested against semantic communication in [24], whose authors consider the integrity of messages and the reliability of the application as dual objectives. More complete threat models for semantic communication are given in [25], [26], which include attacks against various components of the system, including the training process. We observe that these previous works focus on securing the content of the current semantic message, without considering previous transmissions [27]. In addition, side channel attacks, such as the one considered in this work, have been mostly neglected by the semantic communication literature. This is a critical issue, as this type of attack is effective even if the content of

messages is perfectly secure (e.g., with the use of one-time pads).

Interestingly, side channel attacks have been considered in other fields, such as Cloud scheduling. For example, the work in [28] analyzes a model in which a scheduler sends computing jobs to servers to satisfy clients with different arrival times. In this scenario, a malicious entity can infer the traffic patterns of legitimate users by measuring the scheduler's response time. A possible defense is the partial randomization of task execution times [29], which significantly reduces leakage through the side channel at the cost of lower system efficiency. Similar considerations were applied to the field of Information-Centric Networking (ICN), in which caching is used to infer information about the popularity of content [30].

Finally, we consider related work of remote estimation and control field: studies from this area are not closely related to semantic communication and GoC, but they approach similar problems from another angle, and some of their conclusions can be applied to the scenarios studied in this manuscript. In the remote estimation scenario, the secrecy of monitoring systems against side channel attacks is closely related to the concept of *opacity*. In the estimation literature, a system is considered opaque if an eavesdropper with limited observations is unable to estimate some restricted information [31], including the identity of a client or whether the system enters a set of secret states. The analysis of opacity has been extended to K-step observations [32] and scenarios in which the eavesdropper has access to the entire observation history [33]. In information-theoretic terms, opacity can be defined as the difference between the entropy of the belief distributions of the legitimate monitor and the eavesdropper [34].

In control scenarios, where the legitimate agent can affect the evolution of the system through actions, but the action policy is known to the eavesdropper, opacity is more difficult to achieve, and its formal verification becomes a highly complex [35] or even undecidable problem [36]. At the same time, the ability to affect the state of the system enables agents to actively improve security by inserting fictitious events [37] to confuse eavesdroppers. This inherent complexity makes it critical to design GoC policies that optimize control performance under opacity constraints. To the best of our knowledge, the current literature considers the problem of maximizing the opacity of the initial state or the current state, while this work considers the opacity of the entire system history, which is significantly more challenging.

III. Goal-Oriented Communication Model

We consider a remote control scenario in which one node (Alice) can instantaneously observe the state of a discrete-time Markov chain defined by a state space $\mathcal{S}$ and a transition matrix $\mathbf{P}$, whose element $P(s'|s,a)$ corresponds to the probability of moving from state s to state s' after taking action a. We denote by $s(n) \in \mathcal{S}$ the state of the process at time step n and by $\boldsymbol{\mu}_0$ the initial probability distribution of the state. A second node (Bob) is assigned the task of controlling or estimating the process (depending on the scenario considered) by choosing an action over an action space $\mathcal{A}$. We assume that both Alice and Bob know $\mathbf{P}$ and $\boldsymbol{\mu}_0$, but Bob cannot observe $s(n)$ directly and must rely on Alice's transmissions to update his information about the current process state. This assumption is shared with our previous work [14] and is commonly adopted in control scenarios. In practical environments, Alice and Bob may need to spend a short training phase during which they learn the environment model [38], but may then proceed with our assumption in stationary scenarios. The notation used in our model is summarized in Table I.

TABLE I
Model Notation

Symbol	Description	Symbol	Description
$\mathcal{S}$	State space	$\mathcal{A}$	Action space
$\mathbf{P}$	Transition prob. matrix	γ	Discount factor
$r_A(c)$	Alice's reward	$r_B(s,a)$	Bob's reward
$R(s,c,a)$	Total reward	$T_{\max}$	Max time interval
$\boldsymbol{\mu}$	Steady-state distribution	β	Transmission cost
Δ	Steps since last update	τ	Timing signal
$\zeta_{\Delta,s}(s')$	Δ-step belief	$\delta(\cdot,\cdot)$	Kronecker delta
$\sigma(\tau\|s)$	Inter-transmission PMF	$H(\cdot)$	Entropy function
$\pi_c(s,\Delta)$	Communication policy	$\pi_a(s,\Delta)$	Action policy
D	Opacity time gap	$L_{\min}$	Minimum leakage
$L_E(n;D)$	Information leakage	$\boldsymbol{\phi}_E$	Eve's belief
$b_k(s;n)$	Backward probability	$f_k(s)$	Forward probability
η	Eve's estimate	L_{low}	ADE lower threshold
L_{high}	ADE higher threshold	H^*	PDE target entropy
$\xi_{\pi_c}^{(s^*,\tau)}(s,\Delta)$	Single-state deviation	ν	SDE temperature

We consider a *pull-based* configuration in which, at each time step n, Bob must decide whether to ask Alice for an update or to estimate the current state of the Markov chain from the information he already knows. Each transmission incurs a communication cost of some type, which we generally indicate as $\beta \in \mathbb{R}^+$. We denote Bob's binary communication decision as $c(n) \in \{0,1\}$, with $c(n) = 1$ in the case of transmission, and $c(n) = 0$ otherwise. Moreover, we assume a maximum number of steps, $T_{\max}$, after which Bob always requests an update, to preserve tractability. Its impact can be minimized by setting a large value $T_{\max}$.

We assume that the communication delay is shorter than the time step of the underlying Markov process representing the target application. This assumption relies on the time-scale separation between the application layer and the physical layer, ensuring that when Alice transmits, Bob receives the state information within the same time slot. Using the state updates from Alice and his knowledge of $\mathbf{P}$, Bob keeps a local estimate of the state probability distribution of the remote process, that is, a *belief* on the process state that we denote as $\boldsymbol{\zeta}$. Since each transmission represents a renewal of Bob's beliefs, his belief over the process state only depends on the last received state s and the time Δ since the last update [14].

We then define an *action policy* $\pi_a : \mathcal{S} \times \{0,\dots,T_{\max}\} \to \mathcal{P}(\mathcal{A})$, mapping the possible states to a probability space over $\mathcal{A}$, and a *communication policy* $\pi_c : \mathcal{S} \times \{0,\dots,T_{\max}\} \to [0,1]$. Let $\zeta_{\Delta,s}(s')$ represent Bob's estimate of the probability that the process will be in state s' in Δ steps, given that Alice just reported that the process was in state s. This probability can be computed recursively as

$$\zeta_{\Delta,s}(s') = \sum_{s''\in\mathcal{S}} P(s'|s'';\pi_a)\zeta_{\Delta-1,s}(s'')\,, \tag{1}$$

where $\zeta_{0,s}(s') = \delta(s,s')$, and $\delta(m,n)$ is the Kronecker delta function. Bob's action policy π_a is included because,

in the control scenario, the evolution of the Markov process is affected by Bob's actions. In the estimation case, we can simplify (1) to $\zeta_{\Delta,s}(s') = \mathbf{P}^{\Delta}(s,s')$, i.e., to the element with indices s and s' of the Δ-th power of the transition matrix, which does not depend on the action policy π_a.

We then define a *task reward function* $r_B : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that determines the performance of Bob's task (estimation or control). We remark that, when considering remote estimation scenarios, Bob's action consists of estimating the state from the available information, that is, $a(n) = \hat{s}(n)$. The action space then corresponds to the state space, and the transition probabilities of the Markov process are independent of the selected action, i.e., $P(s'|s,a) = P(s'|s,a') \ \forall \ a,a' \in \mathcal{A}$. Hence, the task reward is equal to 1 if the state estimate matches the actual state, and 0 otherwise, i.e., $r_B(s,\hat{s}) = \delta(s,\hat{s})$.

We also introduce Alice's *communication reward function* $r_A : \{0,1\} \to \mathbb{R}$, with $r_A(c) = -\beta c$, where β is a cost that is paid only when Bob asks for a transmission ($c = 1$). The total reward is then given by the combination of the task reward and the (negative or null) communication reward:

$$R(s,c,a) = r_B(s,a) + r_A(c). \tag{2}$$

Bob's goal is to find the communication and action policies that jointly maximize the expected cumulative reward

$$G(\pi_c,\pi_a) = \mathbb{E}\left[\sum_{m=n}^{+\infty} \gamma^{(m-n)} R(s(m),c(m),a(m))\right], \tag{3}$$

where $\gamma \in [0,1)$ is the discount factor and $\mathbb{E}[\cdot]$ denotes the expectation operator. The described problem is a remote Partially Observable Markov Decision Process (POMDP) [39], [40], characterized by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{P}, r_B, \gamma, T_{\max}, \beta \rangle$. The Modified Policy Iteration (MPI) scheme given in [14, Alg. 1] can find the jointly optimal GoC policy in polynomial time with respect to the size $|\mathcal{S}|$ of the state space.

The time between consecutive transmissions, which is denoted as τ, is then a random variable depending on the state s, and its Probability Mass Function (PMF) $\sigma(\tau|s)$ is

$$\begin{aligned}\sigma(\tau|s) &= \text{Prob}(c_\tau = 1, c_\Delta = 0 \,\forall\, \Delta \in \{1,\dots,\tau-1\} | s_0 = s) \\ &= \pi_c(s,\tau) \prod_{\Delta=1}^{\tau-1} (1 - \pi_c(s,\Delta)).\end{aligned} \tag{4}$$

Any time Bob receives an update from Alice, he can determine his future control actions in advance, as well as the timing of the next update request.

IV. Eavesdropping Attack

We assume that an eavesdropper (Eve) knows the Markov process statistics represented by $\mathbf{P}$ and $\boldsymbol{\mu}_0$, and Bob's action and communication policies π_a and π_c. However, Eve cannot directly observe the process, nor read the content of Alice's transmissions. Therefore, she tries to gain information about the state of the Markov chain by observing the intervals between consecutive Bob's requests. From Eve's perspective, the system is a *Hidden Markov Model (HMM)*, where the timing signals τ, i.e., the intervals between consecutive state updates, are the observations from which she determines the maximum a posteriori (MAP) estimate of the Markov source state. The overall scenario is schematized in Fig. 1.

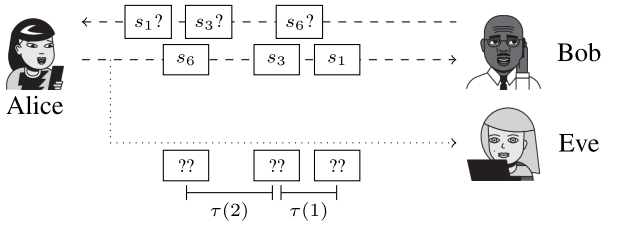


Fig. 1. The goal-oriented eavesdropping attack: Eve cannot decipher Alice's updates, but the timing signal τ allows her to estimate the state of the process.

A. Information Leakage

We now consider the problem of preventing Eve from acquiring information about the remote process state from Alice and Bob's communication.[1] We define the secrecy objective using the concept of the *opacity time gap*, denoted as D. This gap represents the number of past time steps for which information on the Markov chain should remain undisclosed. Let $\boldsymbol{\phi}_E(n;d)$ denote Eve's belief distribution about the process state at time $n-d$, given that she has listened to the channel up to time n. Therefore, $[\boldsymbol{\phi}_E(n;d)](s)$ is the probability that $s(n-d) = s \in S$ from Eve's perspective.

We define the *information leakage* at time step n as

$$L_E(n;D) = \max_{d\in\{0,\dots,D\}} \left\{1 - \frac{H\left(\boldsymbol{\phi}_E(n;d)\right)}{H_0}\right\}, \tag{5}$$

where $H(\cdot)$ is the Shannon information theoretic entropy defined as $H(\mathbf{p}) = -\sum_{s\in\mathcal{S}} p(s)\log_2(p(s))$, where $p(\cdot)$ denotes the PMF of the process state [1]. The denominator $H_0 = \log_2(|\mathcal{S}|)$ is a normalization constant. We note that $L_E(n;D) = 0$ only if $\boldsymbol{\phi}_E(n;d)$ is uniform in the state space for any delay $d \in \{0,\dots,D\}$, which means that Eve does not have information on the state of the system in the last D steps. On the other hand, $L_E(n;D) = 1$ if $\boldsymbol{\phi}_E(n;D) = \delta(s,s_{n-d})$ for some $d \le D$, i.e., Eve has perfect knowledge of the state of the system in at least one of the last D steps. On the other hand, Eve is always able to determine the steady-state distribution $\boldsymbol{\mu}$. The leakage can never be less than

$$L_{\min} = 1 - (H_0)^{-1} H(\boldsymbol{\mu}) \ge 0\,. \tag{6}$$

Zero leakage can thus be achieved only for processes with a uniform steady-state distribution $\boldsymbol{\mu}$, with $H(\boldsymbol{\mu}) = H_0$, whereas $L_{\min} > 0$ in all other cases. Eve's best estimate of state $s(n)$ is obtained at step $n+D$, as she has more observations to draw on. The accuracy of this estimate is

$$\eta(n) = \delta\left(s(n), \arg\max_{s'\in\mathcal{S}} [\boldsymbol{\phi}_E(n+D;D)](s')\right). \tag{7}$$

This setup gives Eve an advantage, as she can wait up to D steps before estimating the process state, while Bob's estimation must be performed in a timely fashion.

[1] We specify that our formulation does not consider Eve's initial knowledge over the Markov source. If the initial state distribution is low-entropy, the mixing time of the chain might be quite long, which leads to an edge case whose analysis is left to future work.

B. Forward-Backward State Estimation

Since Eve sees the system as an HMM, her belief over the process state can be computed through the forward-backward algorithm. Eve combines forward state transition probabilities, which only consider the past, with backward state transition probabilities, which only consider the future. When estimating the state at time m using information up to time $n > m$, the forward probabilities are based on observations up to time m, while the backward probabilities are based on those from time $m+1$ to time n.

Upon observing the k-th request from Bob, Eve can compute the forward probability for any initial state s as

$$f_k(s) = \sum_{s' \in \mathcal{S}} \zeta_{\tau(k),s}(s') \sigma(\tau(k)|s') f_{k-1}(s'), \tag{8}$$

where $\tau(k)$ is the number of steps between transmissions $k-1$ and k, $\boldsymbol{\zeta}_{\Delta,s}$ is given by (1), and $\sigma(\tau|s)$ is given by (4). Recursion starts setting the initial probability vector $\mathbf{f}_0$ equal to the steady-state probability distribution, i.e., $\mathbf{f}_0 = \boldsymbol{\mu}_0$.

The backward probability for the same state is instead

$$b_k(s;n) = \sigma(\tau(k+1)|s) \sum_{s' \in \mathcal{S}} \zeta_{\tau(k+1),s}(s') b_{k+1}(s';n). \tag{9}$$

The last step in the recursive calculation uses $b_{K(n)}(s) = |\mathcal{S}|^{-1} \, \forall s \in \mathcal{S}$, as Eve has no information after index $K(n)$, which represents the index of the last transmission before time step n. Eve's belief over the process status when the k-th update is transmitted is then

$$\phi_k(s;n) = \frac{f_k(s) b_k(s;n)}{\sum_{s' \in \mathcal{S}} f_k(s') b_k(s';n)}. \tag{10}$$

Eve can also compute the belief distribution of the process status ℓ steps after the k-th transmission step as

$$\phi_k^{(\ell)}(s;n) = \sum_{s',s'' \in \mathcal{S}} \phi_k(s';n) \phi_{k+1}(s'';n) \zeta_{\ell,s'}(s) \zeta_{\tau(k+1)-\ell,s}(s''). \tag{11}$$

Using the above formulas, Eve can compute the belief of the state distribution $\boldsymbol{\phi}_E(n;d)$ for any time step n and delay d. The MAP estimate is simply given by the highest-probability state in the belief distribution. The running time of the forward-backward algorithm is $O(|\mathcal{S}|^2 n)$. We can limit n to the mixing time of the Markov chain, which we denote as M.

V. Eavesdropping Defenses

While Bob aims to accurately estimate or control the process, limiting as much as possible the leakage of information, Eve is a purely adversarial attacker who tries to estimate the state of the remote Markov process by exploiting the correlation between the state transitions of the system and the timing between Alice's transmissions. The performance of the system can be defined as the weighted difference between the overall reward and the information leakage, i.e.,

$$\mathbb{E}\left[\sum_{n=0}^{\infty} R(s(n), c(n), a(n)) - \varepsilon L_E(n; D)\right], \tag{12}$$

where $\varepsilon > 0$ is a parameter that can be used to adjust the relative importance of information leakage with respect to Bob's reward. Therefore, Bob aims to find a communication policy to maximize (12), while considering that Eve can use the forward-backward algorithm to update her estimate.

We can model the overall system as a zero-sum *one-sided partially observable stochastic game (OPOSG)* [41]. The solution for the game is a Nash Equilibrium (NE) where any unilateral deviation from a player's policy results in a decrease in that player's performance. Recently, many methods have been proposed to solve the zero-sum OPOSGs, based on the convexity property of the value function [41] or on dividing the problem into sub-games with limited trajectories [42]. However, finding the optimal strategy for Bob requires a computational complexity that grows exponentially with the state space size, as proven by the following theorem from well-known results in game theory.

Theorem 1: The computational time to find the NE of the zero-sum game between Bob and Eve grows exponentially with the size $|\mathcal{S}|$ of the state space.

Proof: A classical result by Dantzig [43] proves that a two-player zero-sum game with payoff matrix $\mathbf{M}$ is equivalent to the following linear programming problem:

$$\text{minimize} \sum_i \mathbf{x} \quad \text{such that } \mathbf{x} \geq 0, \ \mathbf{M}\mathbf{x} \geq 1. \tag{13}$$

In our case, the optimization vector $\mathbf{x}$ represents the probability of selecting each strategy, and its dimension is thus the same as the space of possible communication and action policies that Bob can adopt. Eve then adopts the optimal eavesdropping policy for each of Bob's policies, and each element of $\mathbf{M}$ is the reward from (12) corresponding to the combination of the two policies. Normalizing $\mathbf{x}$ returns the optimal mixed strategy for one of the players. Notably, the size of $\mathbf{x}$ corresponds to the size of the policy space and grows at least exponentially with the number of states $|\mathcal{S}|$. As linear programming problems cannot be solved in logarithmic time with respect to the size of the optimization vector, the zero-sum game is unsolvable in polynomial time. ■

Although finding an NE is computationally intractable for nontrivial problem sizes, we can design simple heuristic policies that allow Bob to trade-off between communication efficiency and system secrecy, reducing the vulnerability of GoC strategies to timing attacks. In the following, we propose three solutions to attain this objective: the Alternating Defense from Eavesdropping (ADE), which alternates between GoC and periodic transmission, the Packing Defense from Eavesdropping (PDE), which is designed to reduce the entropy of π_c, and the Stochastic Defense from Eavesdropping (SDE), which increases the randomness of communication decisions associated with the optimal GoC policy.

A. Alternating Defense

We know that the optimal GoC communication policy outperforms the optimal Periodic Policy (PP) in terms of expected reward, i.e., it can obtain the minimum transmission cost for a given state-estimation accuracy [14, Th. 2]. However, GoC is highly vulnerable to timing attacks, while a periodic strategy minimizes information leakage, as we prove below.

Theorem 2: In an estimation scenario over a recurrent Markov chain, any periodic scheduling is perfectly private, i.e., the information leakage tends to the minimum value $L_{\min}$ as n increases for any finite value of D.

Proof: Under a periodic policy with period T, we have $\sigma(\tau|s) = \delta(\tau, T)\ \forall\, s \in \mathcal{S}$ and, consequently, the forward probabilities are $f_k(s) = \sum_{s' \in \mathcal{S}} \left(\mathbf{P}^T\right)_{s',s} f_{k-1}(s')$. This is exactly equivalent to a blind update, and the same holds for the backward probabilities. As timing does not provide new information, Eve's belief tends to the steady-state distribution $\boldsymbol{\mu}$ for any n larger than the system mixing time, reducing the leakage to $L_{\min}$, defined in (6), as the window for the leakage calculation moves past the initial transient. ■

We note that the theorem may not always hold in the more general control case, as Bob's action policy π_a affects the steady-state distribution $\boldsymbol{\mu}$. However, the general principle holds, as periodic transmission strategies still minimize leakage for any sequence of control decisions.

Algorithm 1 Alternating Defense from Eavesdropping (ADE)

function SCHEDULE($s, \pi_c^{\text{MPI}}, \pi_c^{\text{PP}}, \mathbf{P}, \mathbf{f}, \mathbf{b}, \boldsymbol{\tau}, L_{\text{low}}, L_{\text{high}}, \alpha$)
  if $\alpha = 0$ **then** ▷ GoC active
    if $L_E(\pi_c^{\text{MPI}}) \geq L_{\text{high}}$ **then** ▷ Check secrecy threshold
      return $\tau = T$, $\alpha = 1$ ▷ Switch to PP
    else
      return next τ following $\sigma^{\text{MPI}}(\tau|s)$, $\alpha = 0$ ▷ Keep using GoC
  else ▷ PP active
    if $L_E(\pi_c^{\text{PP}}) < L_{\text{low}}$ **then** ▷ Check performance threshold
      return next τ following $\sigma^{\text{MPI}}(\tau|s)$, $\alpha = 0$ ▷ Switch to GoC
    else
      return $\tau = T$, $\alpha = 1$ ▷ Keep using PP
end function

We take advantage of this principle to design our first heuristic policy, Alternating Defense from Eavesdropping (ADE), whose pseudocode is reported as Algorithm 1. We maintain a parameter α, which acts as a switch: Bob starts each episode with $\alpha = 0$, i.e., using MPI. As Bob knows his own transmission policy and, hence, the timing signal observed by Eve, he can compute the information leakage during the next transmission interval. Hence, he can set $\alpha = 1$ and switch to a Periodic Policy (PP) whenever the expectation of future leakage increases beyond an upper threshold L_{high} and switch back to GoC by returning to $\alpha = 0$ whenever the future leakage goes below a threshold L_{low}.

This hysteresis pattern allows Bob to limit both the average and maximum leakage, while still exploiting GoC at least in some time intervals. The minimum and maximum leakage will depend on the specific task. These parameters must be set empirically, depending on the leakage and task performance requirements, under the constraints that they should both be in the interval $(H_0, 1)$, as they should be achievable leakage values, and that $L_{\text{high}} > L_{\text{low}}$. This represents a weakness of the Alternating Defense from Eavesdropping (ADE) scheme. Another weakness is its runtime computational complexity, as Bob needs to maintain the same HMM model as Eve, incurring an $O(|\mathcal{S}|^2 M)$ computational cost at each step, where M is the mixing time of the underlying Markov chain.

B. Packing Defense

The second heuristic policy is named Packing Defense from Eavesdropping (PDE) and is based on a simple observation: if multiple states are mapped to the same inter-transmission period, the leakage of the timing signal decreases, as Eve has a harder time distinguishing between states. We then define the entropy of the communication policy π_c as

$$H(\pi_c) = -\sum_{\tau=1}^{T_{\max}} \frac{\sum_{s \in \mathcal{S}} \sigma(\tau|s)}{|\mathcal{S}|} \log_2 \left(\frac{\sum_{s \in \mathcal{S}} \sigma(\tau|s)}{|\mathcal{S}|} \right). \quad (14)$$

Notably, $\sigma(\tau|s) \in \{0, 1\}$, since PDE is a deterministic policy, and so is the optimal GoC policy returned by MPI.

We can assume that $H(\pi_c)$ is a good proxy for leakage: any periodic policy has zero entropy, while the maximum entropy $\log_2(|\mathcal{S}|)$ is achieved by picking a different inter-transmission interval for each state. In this case, any timing signal τ is mapped to a different state, so that at each transmission Eve gains perfect knowledge of the transmitted value.

Algorithm 2 Packing Defense from Eavesdropping (PDE)

function PACK(π_c, H^*)
  $H \leftarrow$ENTROPY(π_c) ▷ Compute entropy using (14)
  running $\leftarrow$ true
  while running **do**
    running $\leftarrow$ false, $R \leftarrow -\infty$, $\pi_c' \leftarrow \pi_c$
    for $s^* \in \mathcal{S}$ **do**
      for $\tau \in \{1, \ldots, T_{\max}\}$ **do**
        if ENTROPY($\xi_{\pi_c}^{(s^*,\tau)}$)$< H$ **then** ▷ Check entropy
          if REWARD($\xi_{\pi_c}^{(s^*,\tau)}$)$> R$ **then**
            $\pi_c' \leftarrow \xi_{\pi_c}^{(s^*,\tau)}$
            $R \leftarrow$REWARD(π_c')
    if $\pi_c' \neq \pi_c$ **then**
      $\pi_c \leftarrow \pi_c'$ ▷Update policy
      $H \leftarrow$ENTROPY(π_c)
      if $H > H^*$ **then** ▷ Stopping criterion
        running $\leftarrow$ true
end function

To define the PDE strategy, we introduce the concept of *single-state deviation policy* $\xi_{\pi_c}^{(s^*,\tau)}$, which is a scheduling strategy identical to π_c except for state s^*, whose associated scheduling period is set to τ:

$$\xi_{\pi_c}^{(s^*,\tau)}(s, \Delta) = \delta(\Delta, \tau)\delta(s, s^*) + \pi_c(s, \Delta)(1 - \delta(s, s^*)). \quad (15)$$

Starting from the purely goal-oriented policy, denoted by $\pi_c^{(0)}$, we can then define an iterative procedure to *pack* the policy through a series of single-state deviations that gradually reduce the entropy. The i-th packing iteration is defined as $\pi_c^{(i)}(s, \Delta) = \xi_{\pi_c^{(i-1)}}^{(s_i^*,\tau_i)}(s, \Delta)$ for all $s \in S$, where

$$(s_i^*, \tau_i) = \underset{(s^*,\tau): H\left(\xi_{\pi_c^{(i-1)}}^{(s^*,\tau)}\right) < H(\pi_c^{(i-1)})}{\arg\max} G\left(\xi_{\pi_c^{(i-1)}}^{(s^*,\tau)}\right). \quad (16)$$

This packing rule ensures that the new policy $\pi_c^{(i)}$ is the one that maximizes the expected long-term reward G among those with entropy lower than $H\left(\pi_c^{(i-1)}\right)$. We can repeat the packing step until the final policy achieves a target entropy

value H^*, which represents the stopping criterion for PDE. The full PDE pseudocode is given in Algorithm 2.

Since the PDE defense directly changes Bob's policy, its runtime complexity is $O(1)$, assuming that the policy is stored in a random access memory. The additional time required for the offline policy computation is asymptotically dominated by the time required by MPI, which is $O(|\mathcal{S}|^4 T_{\max}^3 |\mathcal{A}|)$: each new single-deviation policy requires iterating over all states and potential inter-transmission time, with complexity $O(|\mathcal{S}|T_{\max})$, and states can be packed up to $|\mathcal{S}|$ times before collapsing into the periodic policy. The overall added offline computational complexity is then $O(|\mathcal{S}|^2 T_{\max})$. PDE can then be run without a significant computational burden, at the expense of a slightly longer offline policy computing phase.

C. Stochastic Defense

The third defense scheme, named SDE, takes advantage of the same intuition at the basis of PDE: a higher entropy $H(\pi_c)$ of the communication policy leads to a reduced information leakage for the whole system. We consider the optimal GoC strategy given by the MPI algorithm and build a stochastic variation of the associated communication strategy π_c^{MPI}.

The MPI algorithm computes, for each state-action pair $(s, \Delta, c) \in \mathcal{S} \times \mathbb{Z}^+ \times \{0, 1\}$ a Q-value $Q_{\text{MPI}}(s, \Delta, c)$ representing the the expected long-term reward associated with the MPI policy. In particular, $Q_{\text{MPI}}(s, \Delta, 0)$ represents the benefit of transmitting a packet after receiving state s and awaiting Δ slots, while $Q_{\text{MPI}}(s, \Delta, 1)$ represents the benefit of staying silent in the same circumstances. Given a state (s, Δ), the optimal communication decision is then $c = \arg\max_{\{0,1\}} \{Q_{\text{MPI}}(s, \Delta, c)\}$. This means that the policy π_c^{MPI} is deterministic, and each state s is univocally associated to an inter-transmission interval Δ_s.

Instead of following optimal communication decisions, SDE associates each state (s, Δ) with probability values, making the communication policy π_c stochastic. In particular, the probability $\pi_c(s, \Delta)$ of triggering a new state transmission after receiving state s and waiting Δ slots is given by the softmax function computed over the Q-value distribution:

$$\pi_c^{\text{SDE}}(s, \Delta) = \frac{e^{Q_{\text{MPI}}(s,\Delta,1)/\nu}}{\sum_{g\in\{0,1\}} e^{Q_{\text{MPI}}(s,\Delta,g)/\nu}}, \tag{17}$$

where $\nu \in \mathbb{R}^+$ is a temperature controlling the randomness of the communication policy. As $\nu \to 0$, π_c^{SDE} becomes closer to the deterministic policy π_c^{MPI}, while higher values of ν increase the randomness of the scheduling, making the policy closer to a random coin flip. In the extreme case with $\nu \to +\infty$, we will have $\pi_c(s, \Delta) = \frac{1}{2}\, \forall s, \Delta$ and $\sigma(\tau|s) = 2^{-\Delta}$.

On a practical level, the SDE policy makes the agent's action more unpredictable, increasing $H(\pi_c)$ and, consequently, reducing the leakage of the system. From a computational perspective, SDE requires computing the optimal GoC policy through the MPI algorithm. Once the temperature ν has been set, the action policy π_a must be re-computed in order to adapt the control actions to the modified communication policy. Hence, tuning the defense scheme requires an additional cost $O(|\mathcal{S}|T_{\max})$. On the other hand, the runtime complexity of SDE is $O(|\mathcal{A}|)$ if the Q-values can be stored as a table. The full policy calculation and runtime computational costs of all considered schemes are listed in Table II.

TABLE II
COMPUTATIONAL COST OF THE PROPOSED SCHEMES

Scheme	Policy (offline)	Defense (offline)	Runtime
PP	$O(T_{\max})$ [14]	0	$O(1)$
MPI	$O(\|\mathcal{S}\|^4 T_{\max}^3 \|\mathcal{A}\|)$ [14]	0	$O(1)$
ADE	$O(\|\mathcal{S}\|^4 T_{\max}^3 \|\mathcal{A}\|)$	$O(1)$	$O(\|\mathcal{S}\|^2 M)$
PDE	$O(\|\mathcal{S}\|^4 T_{\max}^3 \|\mathcal{A}\|)$	$O(\|\mathcal{S}\|^2 T_{\max})$	$O(1)$
SDE	$O(\|\mathcal{S}\|^4 T_{\max}^3 \|\mathcal{A}\|)$	$O(1)$	$O(\|\mathcal{A}\|)$

VI. SIMULATION SETTINGS AND RESULTS

In the following, we study our GoC model in two Monte Carlo [44] simulation scenarios. The first represents a *remote estimation* task, where Bob aims to estimate the current state of the system, which evolves independently from Bob's actions. The second is a *remote control* task in which Bob affects the evolution of the system with the goal of reaching certain states. After presenting each scenario, we analyze the performance of the heuristic policies introduced in Sec. V against the optimal GoC scheduling, which is computed via the Modified Policy Iteration (MPI) algorithm, and a Periodic Policy (PP).

A. Scenario Settings

The remote estimation and remote control scenarios are both modeled according to the discrete time POMDP presented in Sec. III. Although the proposed framework is valid for any recurrent Markov chain, we focus on a class of processes that allow for an easy analysis of the system's behavior under different conditions. We consider a state space of $|\mathcal{S}| = 30$ states, numbered from 1 to 30. The transition probability matrix $\boldsymbol{P}$ depends on a single *density decay* parameter θ, that makes it possible to tune the predictability of the evolution of the system. We have

$$P(s, s', a) = \begin{cases} \dfrac{2 - 2\ g(s,\theta)}{6}, & s' = \chi(s,a) \oplus 1,\ \ \mathrm{mod}(s,4) = 2; \\ \dfrac{2 + g(s,\theta)}{6}, & s' = \chi(s,a) \oplus 3,\ \ \mathrm{mod}(s,4) = 2; \\ \dfrac{2 + g(s,\theta)}{6}, & s' = \chi(s,a) \ominus 2,\ \ \mathrm{mod}(s,4) = 2; \\ \dfrac{1 + 2\ g(s,\theta)}{3}, & s' = \chi(s,a) \oplus 1,\ \ \mathrm{mod}(s,4) \neq 2; \\ \dfrac{1 - g(s,\theta)}{3}, & s' = \chi(s,a) \oplus 3,\ \ \mathrm{mod}(s,4) \neq 2; \\ \dfrac{1 - g(s,\theta)}{3}, & s' = \chi(s,a) \ominus 2,\ \ \mathrm{mod}(s,4) \neq 2; \\ 0, & \text{otherwise}; \end{cases} \tag{18}$$

where $\oplus$ and $\ominus$ represent modulo $|\mathcal{S}|$ addition and subtraction, $\mathrm{mod}(m, n)$ is the integer modulo function, $\theta \in \mathbb{R}^+$ is the density decay, and $g(s, \theta)$ is defined as

$$g(s, \theta) = \left|2(s-2)(|\mathcal{S}|-2)^{-1} - 1\right|^{\theta} \in [0, 1]. \tag{19}$$

The function $\chi(s,a) \in \mathcal{S}$ determines the state transition associated with action $a \in \mathcal{A}$, which is $\chi(s,a) = s$ in remote estimation (therefore, independent of Bob's actions), and $\chi(s,a) = s + a$ in the case of remote control.

From any state s, transitions can occur with a non-zero probability to only three landing states that, only for the control scenario, depend on the action a. The probabilities of moving to the farthest reachable states ($\chi(s,a) \oplus 3$ or $\chi(s,a) \ominus 2$) are always balanced. Instead, the transition to the intermediate state $\chi(s,a) \oplus 1$ has a higher probability than the other two transitions from all states, except those such that $\mathrm{mod}(s,4) = 2$, making the drift of the process more variable.

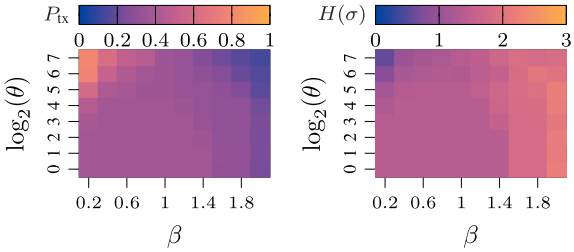
As $\theta \to \infty$, $g(s,\theta)$ tends to zero, and the transition probabilities to neighboring states will become more uniform (and less predictable). Conversely, as $\theta \to 0$, $g(s,\theta)$ tends to 1 and most states will have deterministic and fully predictable transitions. Finally, $g(s,\theta) = 1$ for the extreme states $s = 1$ and $s = |S|$, progressively decreasing when moving towards the middle states. For any value of θ, middle states tend to have more balanced transition probabilities toward their landing states, while states closer to the extremes have more unbalanced transition probabilities, i.e., more predictable transitions.

As already mentioned, Bob's action space $\mathcal{A}$ in the estimation scenario is identical to the state space, and the task reward function is $r_B(s,a) = \delta(s,a)$. In the remote control scenario, the action space is $\mathcal{A} = \{0,1,2\}$ and we defined $\chi(s,a) = s + a$. Therefore, Bob can (stochastically) control the sequence of states by choosing proper actions. In our experiments, we assumed the control goal was to keep the remote process close to the middle state $s^\circ = 14$. Accordingly, we define the reward as $r_B(s,a) = 5 \cdot \exp(-|s - s^\circ|)$, $\forall\, s \in \mathcal{S}$. Note that the control reward does not depend on the accuracy of Bob's estimates but only on the distance between the current state s and the target state s°.

In both scenarios, we generate multiple POMDPs by varying the density decay $\theta \in [1, 2^7]$ and the transmission cost $\beta \in [0.2, 2]$. We compute the optimal GoC communication policy given by MPI [14], maximizing theov cumulative reward of Bob penalized by the communication cost, as defined in (3). We compare MPI with PP, which is the best policy among those exploiting a fixed inter-transmission period, and the three heuristics presented in Sec. V. In particular, ADE uses $L_{\text{low}} = 0.4$ and $L_{\text{high}} = 0.6$ as leakage thresholds, while PDE sets $H^* = \frac{1}{2} H\left(\pi_c^{(0)}\right)$ as a stopping criterion, where $\pi_c^{(0)}$ is the initial communication policy returned by the MPI algorithm. Finally, we tune SDE by considering $\nu = 0.3$ as the temperature of the softmax function in (17). In all cases, we set $T_{\max} = 10$ as the maximum interval between consecutive transmissions, setting $\pi_c(s,10) = 1\, \forall s \in \mathcal{S}$.

B. Remote Estimation Scenario

We first consider the remote estimation scenario and analyze the policy returned by the MPI algorithm. Fig. 2(a) shows a heatmap of the transmission probability associated with each configuration of the system. Transmissions become less likely as β increases and are also affected by the randomness of the system's evolution, which depends on the density decay θ. When the transmission cost is low ($\beta \to 0$), larger values of θ (with less predictable transition matrices) result in more frequent update requests from Bob, who can exploit communication to keep track of the process evolution. However, if the transmission cost increases ($\beta \to 2$), the trend reverts and the as probability decreases as θ increases, because the higher estimation accuracy may not balance the update cost.



(a) Transmission probability. (b) Entropy of π_c.

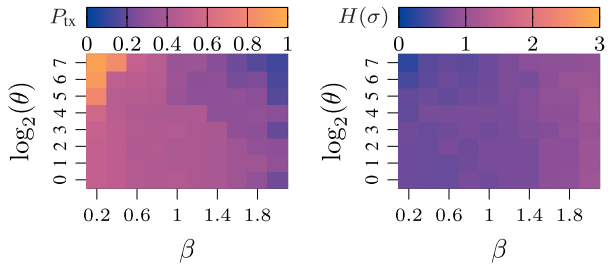
Fig. 2. Characterization of the MPI policy as a function of β and the density decay θ in the estimation scenario.

Fig. 2(b) represents the entropy $H(\pi_c)$ of the communication policy π_c, which, as discussed in Sec. V, is a proxy of information leakage caused by transmission decisions. We observe that $H(\pi_c)$ decreases as $\beta \to 0$: when the frequency of communication increases, the variability of the inter-transmission time decreases, and Eve has more difficulty in sorting out the states sequence from the timing signal. This phenomenon is more evident for $\theta \to 2^7$, which represents a condition in which state transitions are less predictable and the optimal scheduling becomes similar to the PP strategy.

A policy that selects a different inter-transmission time for each state would have an entropy equal to $\log_2(|\mathcal{S}|)$, while any periodic policy would have zero entropy. In general, we expect the MPI algorithm to have a higher entropy than all other strategies analyzed in this paper, with the exception of SDE. This is because PP uses a fixed inter-transmission period, ADE alternates between MPI and PP, while PDE is explicitly designed to reduce $H(\pi_c)$ with respect to MPI. On the other hand, SDE is designed to increase the entropy of Bob's communication decisions in a way that makes state estimation more difficult for Eve. While an increased entropy in deterministic decisions also increases the mutual information between τ and the state, increasing entropy by randomizing decisions reduces the mutual information that Eve can exploit, leading to a more secure communication.

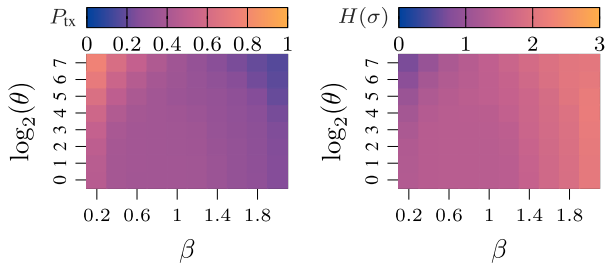
Fig. 3 and Fig. 4, which report the transmission probability and entropy associated with PDE and SDE, respectively, confirm this intuition. PDE's transmission probability is similar to MPI's, but PDE halves the entropy of the communication policy in all configurations of the system. On the other hand, SDE leads to an increased entropy in most scenarios, except in cases in which the MPI policy tends to always transmit the state, i.e., it already has zero entropy.

Fig. 5 shows the information leakage L_E during a single episode of $N_{\text{step}} = 200$ steps, considering $\beta = 1$, $\theta = 32$ as scenario parameters, and $D = 5$ as opacity time gap. We can observe that the leakage of the MPI algorithm quickly approaches 1, showing that Eve correctly guesses the remote



(a) Transmission probability. (b) Entropy of π_c.

Fig. 3. Characterization of the PDE policy as a function of β and the density decay θ in the estimation scenario.



(a) Transmission probability. (b) Entropy of π_c.

Fig. 4. Characterization of the SDE policy as a function of β and the density decay θ in the estimation scenario.

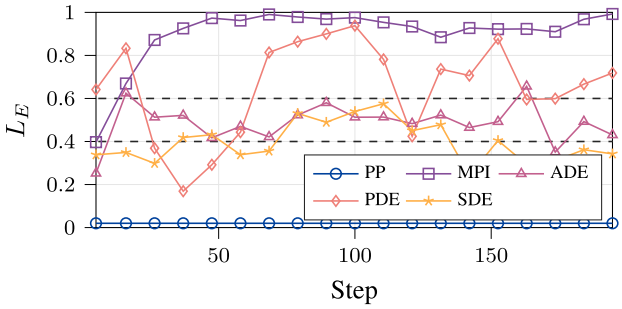
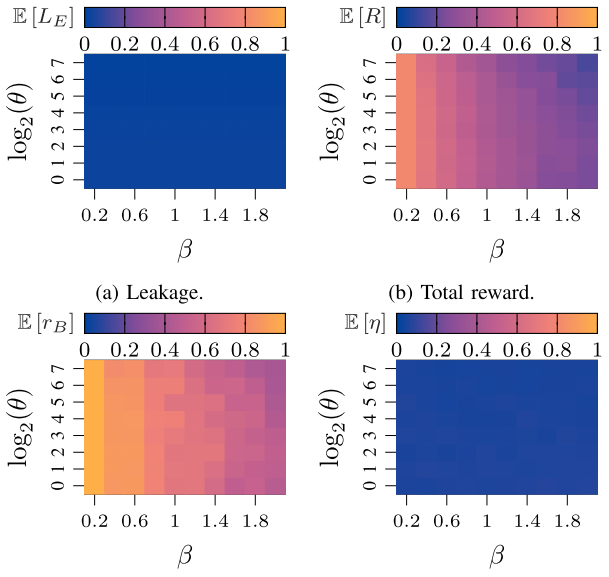


Fig. 5. Information leakage during a single episode in the estimation scenario, with $\beta = 1$, $\theta = 32$ and $D = 5$. The ADE thresholds L_{low} and L_{high} are marked as dashed lines.

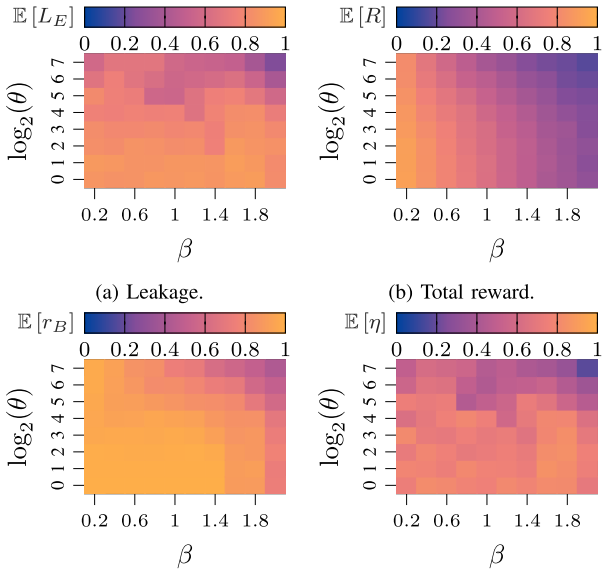
state very frequently. In contrast, PP does not provide any information to Eve, whose knowledge is limited to the steady-state probability distribution of the Markov process. By design, the ADE algorithm keeps the leakage between L_{low} and L_{high}, offering a compromise between the two previous approaches. All defense policies manage to reduce the leakage, but ADE and SDE are able to maintain a more stable leakage, while PDE tends to oscillate more, exposing the system to a high vulnerability in some steps.

Fig. 6 shows the performance of PP while varying the communication cost β and density decay θ, and considering a total of $N_{\text{ep}} = 10$ episodes for each configuration, with $N_{\text{step}} = 200$. The figure shows the leakage, the total reward R, defined in (2), the reward for the estimation task r_B, and the probability η that Eve correctly estimates the state of the Markov process, defined as in (7). We observe that $L_E \approx 0$ for all system configurations, following the result in Theorem 2. On the other hand, Fig. 6(a) and Fig. 6(b) show



(a) Leakage. (b) Total reward.
(c) Estimation reward. (d) Eve's accuracy.

Fig. 6. PP performance as a function of θ and β in the estimation scenario, with $D = 5$.

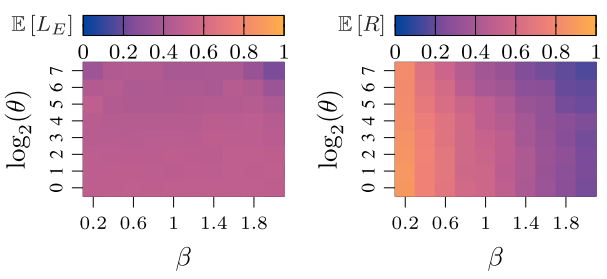


(a) Leakage. (b) Total reward.
(c) Estimation reward. (d) Eve's accuracy.

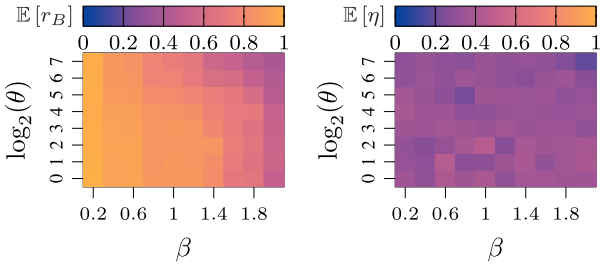
Fig. 7. MPI performance as a function of θ and β in the estimation scenario, with $D = 5$.

that the expected total reward $\mathbb{E}[R]$, as well as Bob's state estimation accuracy r_B, decrease for larger transmission costs (β, which yield longer inter-transmission periods) and more erratic transition probabilities ($\theta \gg 1$).

Fig. 7 offers a comparison of the performance indicators for the MPI strategy, which represents the other extreme, as it is purely oriented towards GoC. Unsurprisingly, this setting leads to a strong secrecy degradation, as shown in Fig. 7(a): the information leakage is close to 0.8 for all configurations except for those with very high values of β and θ. Fig. 7(d) shows that Eve is able to correctly decode the status of the monitored process almost as often as Bob, further highlighting the vulnerability of MPI to timing attacks. On the other hand,

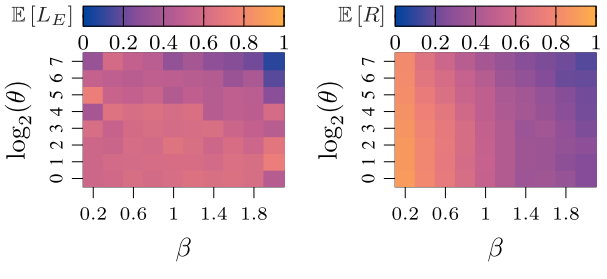


(a) Leakage. (b) Total reward.

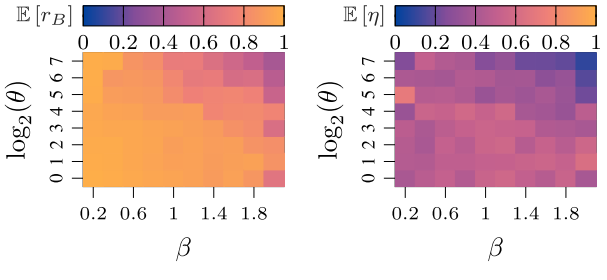


(c) Estimation reward. (d) Eve's accuracy.

Fig. 8. ADE performance as a function of θ and β in the estimation scenario, with $D = 5$.



(a) Leakage. (b) Total reward.

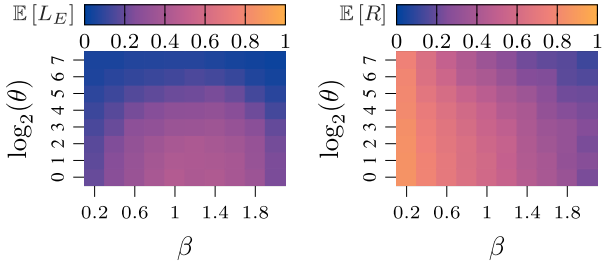


(c) Estimation reward. (d) Eve's accuracy.

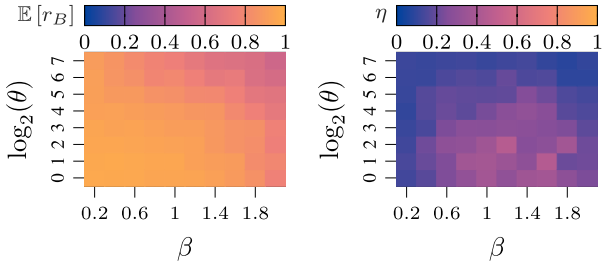
Fig. 9. PDE performance as a function of θ and β in the estimation scenario, with $D = 5$.

MPI significantly improves the total reward compared to PP, as shown in Fig. 7(b). The GoC communication policy tends to transmit more frequently than PP and the gain of MPI over PP reaches 50% when $\beta \to 2$.

In Figs. 8–10, we conduct the same analysis for the three heuristic algorithms. As shown in Fig. 8(a), ADE improves secrecy in all configurations, guaranteeing that the information leakage remains lower than L_{high}. In particular, in this scenario, Eve's accuracy is much lower than Bob's, leading to a mean leakage of 0.45. Fig. 8(b) shows that ADE degrades the total reward compared to MPI, especially in the case of Markov



(a) Leakage. (b) Total reward.



(c) Estimation reward. (d) Eve's accuracy.

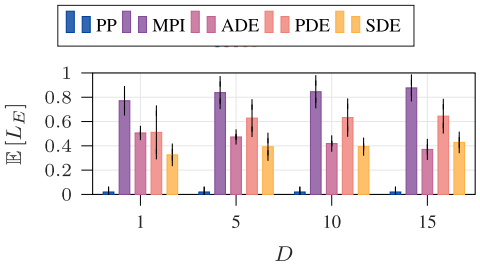
Fig. 10. SDE performance as a function of θ and β in the estimation scenario, with $D = 5$.

chains with low θ and high transmission cost. On the other hand, the reward of ADE presents a performance gain of approximately 10% over PP, as apparent from the comparison between Fig. 8(b) and Fig. 6(b).

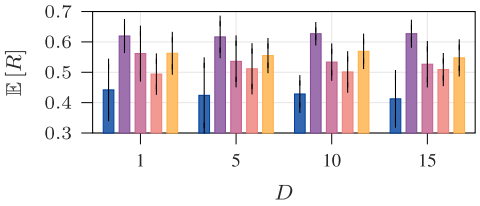
In Fig. 9, we report the results for PDE, which, contrary to ADE, does not explicitly monitor the information leakage but considers $H(\pi_c)$ as a secrecy indicator. As shown in Fig. 9(a), PDE leads to a higher expected leakage than ADE without significantly improving Bob's estimation reward. Hence, ADE performs better than PDE in the estimation task, but the higher computational complexity may make ADE unsuitable for implementation on nodes with limited hardware.

Fig. 10 analyzes the performance of SDE, which markedly reduces the information leakage with respect to all other alternatives, especially for the high randomness ($\theta \to 2^7$) and low communication cost ($\beta \to 0$) scenarios. Looking at Figs. 10(c) and 10(d), we can observe that Eve's estimation accuracy strongly decreases, while the reward associated with Bob's task is not impaired by the defense mechanism. This indicates that SDE constitutes the most effective defense in the estimation task. In general, the trade-off between secrecy and performance is complex, and the defense strategy hyperparameters also need to be appropriately tuned. We will discuss this trade-off more in depth in Sec. VI-D.

Finally, in Fig. 11, we analyze the impact of the time gap D on overall performance, focusing on a system with $\beta = 1$ and $\theta = 32$, and setting $D \in \{1, 5, 10, 15\}$. As expected, the leakage of MPI is an increasing function of D, as a longer opacity gap allows Eve to take advantage of more information to decode the state. As PDE is directly derived from MPI, the two strategies follow a similar trend in terms of information leakage. We also observe that as D increases, ADE adopts increasingly conservative decisions and switches to PP more often and for longer periods. SDE leads to the lowest leakage, while its reward is similar to ADE's and higher than PDE's.

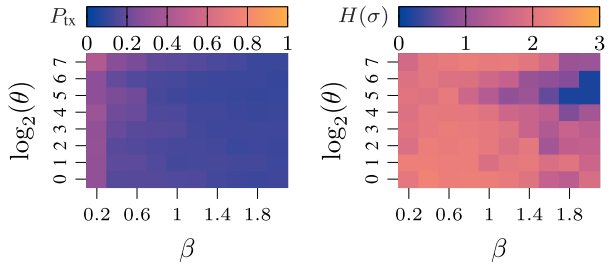


(a) Information leakage.



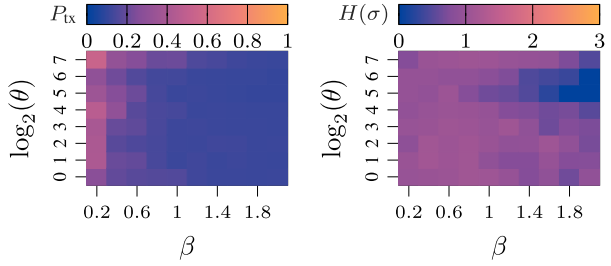
(b) Total reward.

Fig. 11. Expected leakage and reward as a function of D in the estimation scenario, with $\beta = 1$ and $\theta = 32$.



(a) Transmission probability. (b) Entropy of π_c.

Fig. 12. Characterization of the MPI policy as a function of θ and β in the control scenario.

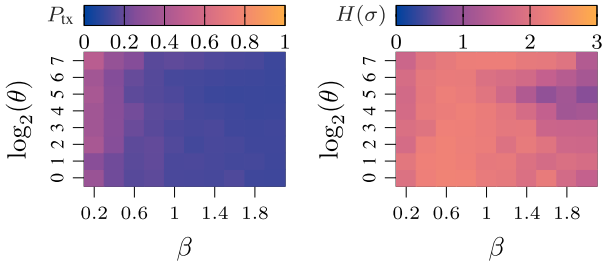


(a) Transmission probability. (b) Entropy of π_c.

Fig. 13. Characterization of the PDE policy as a function of θ and β in the control scenario.

C. Remote Control Scenario

The remote control scenario has a significant difference from the remote estimation scenario: Bob's reward depends on the distance between the current state $s(n)$ and the target state s°, which Bob can partially control. This substantially reduces the transmission probability of the MPI strategy, which, as shown in Fig. 12(a), leads to high data rates only when the Markov process becomes highly stochastic ($\theta \gg 1$) or if the transmission cost is negligible ($\beta \to 0$). A similar trend can be observed for PDE and SDE, whose transmission rates are reported in Fig. 13(a) and Fig. 14(a), respectively.



(a) Transmission probability. (b) Entropy of π_c.

Fig. 14. Characterization of the SDE policy as a function of β and the density decay θ in the control scenario.

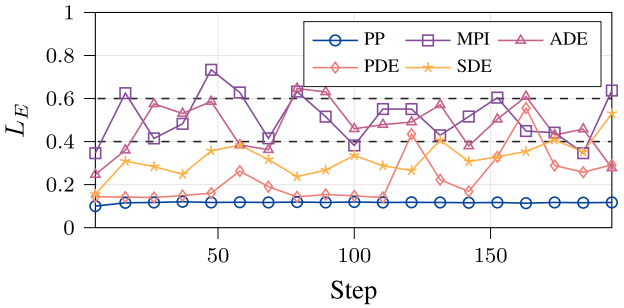
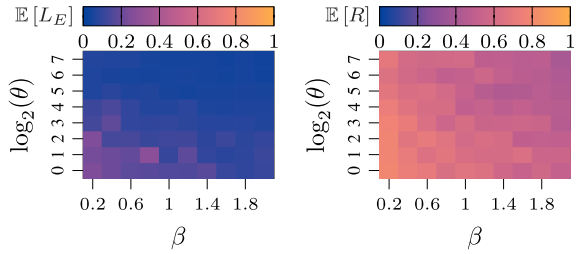


Fig. 15. Information leakage during a single episode in the control scenario, with $\beta = 1$, $\theta = 32$ and $D = 5$. The ADE thresholds L_{low} and L_{high} are marked as dashed lines.

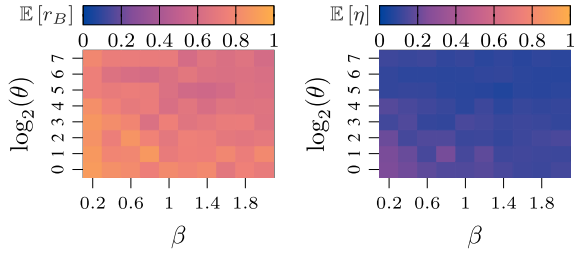
Looking at Fig. 12(b), we can observe how the entropy of the MPI scheduling strongly decreases for $\theta > 16$ and $\beta > 1.4$. In such cases, requesting state updates from Alice is inconvenient and MPI associates most states with the maximum inter-transmission interval $T_{\max}$. On the other hand, the entropy increases again for $\theta > 64$, denoting that the relation between the stochasticity of the system and the corresponding communication policy is governed by many heterogeneous factors. Fig. 13(b) shows that PDE follows the same pattern and reduces the entropy of π_c by 50% in all configurations. As in the estimation case, SDE, whose policy entropy is reported in Fig. 14(b), increases $H(\pi_c)$ by adding randomness to scheduling decisions.

In Fig. 15, we focus on a single episode (with $\theta = 32$, $\beta = 1$, and $D = 5$) and compare the leakage obtained by MPI, PP, and the three heuristic strategies. First, we observe that the leakage of PP has higher values than in the remote estimation task, which denotes that the steady-state distribution $\boldsymbol{\mu}(\pi)$ of the underlying Markov process presents a higher entropy. This is because Bob aims to keep the current state as close as possible to s°, reducing the system's randomness and consequently increasing the information leakage independently of the scheduling decision. On the other hand, the information leakage associated with MPI does not increase beyond 0.6 and ADE rarely switches to periodic communication. Finally, we observe that PDE substantially improves secrecy with respect to both ADE and SDE.

The fact that PP may have a more significant leakage in control tasks is confirmed by Fig. 16(a), which reports $\mathbb{E}[L_E]$ for all the system configurations while using periodic transmissions. Interestingly, PP is more vulnerable to timing attacks for $\beta \to 0$ and $\theta \to 0$, representing the case in which

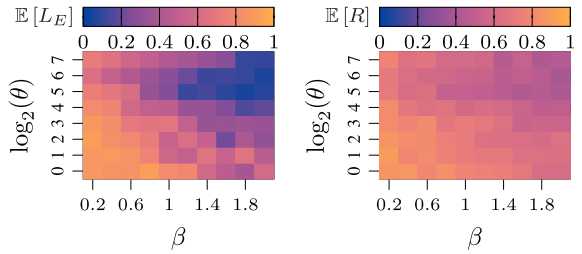


(a) Leakage. (b) Total reward.

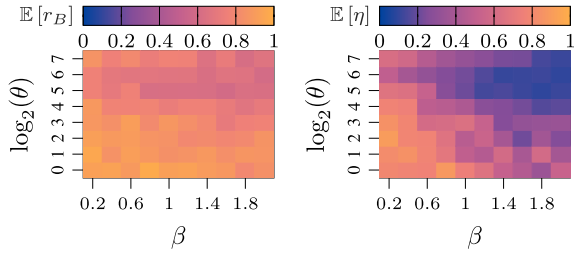


(c) Control reward. (d) Eve's accuracy.

Fig. 16. PP performance as a function of θ and β in the control scenario, with $D = 5$.

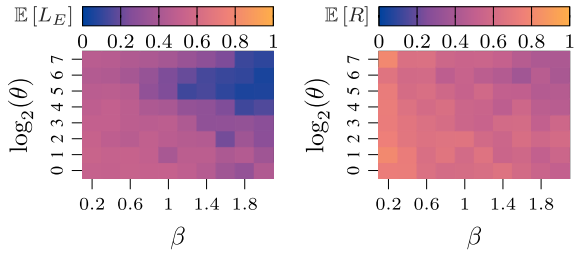


(a) Leakage. (b) Total reward.

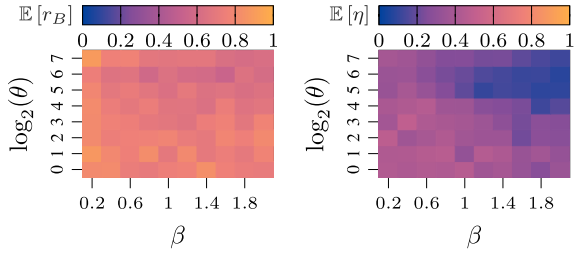


(c) Control reward. (d) Eve's accuracy.

Fig. 17. MPI performance as a function of θ and β in the control scenario, with $D = 5$.

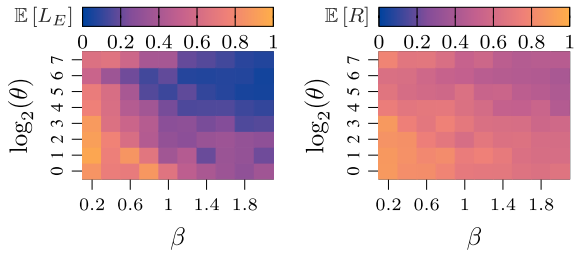


(a) Leakage. (b) Total reward.

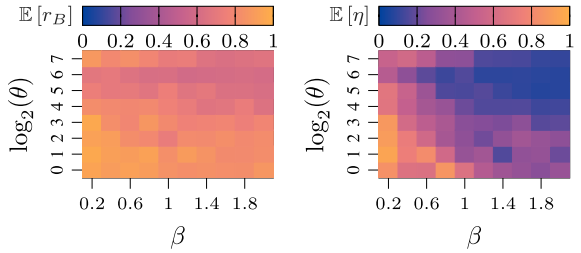


(c) Control reward. (d) Eve's accuracy.

Fig. 18. ADE performance as a function of θ and β in the control scenario, with $D = 5$.



(a) Leakage. (b) Total reward.



(c) Control reward. (d) Eve's accuracy.

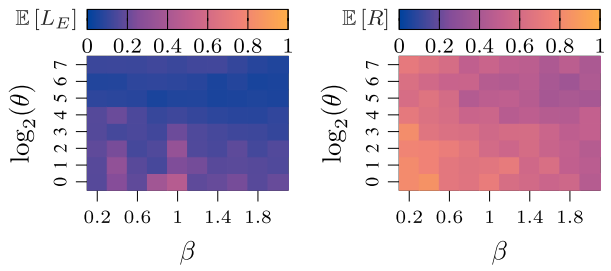
Fig. 19. PDE performance as a function of θ and β in the control scenario, with $D = 5$.

Markov transitions are more deterministic. The same settings lead to an increase in the average reward $\mathbb{E}[r_B]$ of the remote control task and a slight increase in Eve's accuracy $\mathbb{E}[\eta]$.

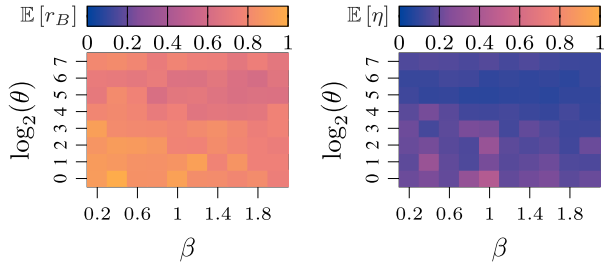
Fig. 17 analyzes the performance of the MPI strategy. Comparing Fig. 17(a) and Fig. 12(b), we observe that the information leakage strongly decreases in the region associated with a low entropy for the scheduling strategy. Looking at Fig. 17(d), we see that this phenomenon also affects Eve's accuracy, as communication secrecy is maximized for $\beta \to 2$ and $\theta \to 2^7$. In particular, improvement in secrecy leads to a reduction in task reward, although this effect is less pronounced than in the remote estimation scenario.

As we can observe from Fig. 18, ADE significantly reduces the accuracy of Eve's estimates, improving the system secrecy, especially when $\beta \to 0$ and $\theta \to 2$. On the other hand, since its goal is to avoid information leakage exceeding L_{high}, ADE continues to use MPI in those scenarios when the goal-oriented communication policy is already opaque. In contrast to ADE, PDE and SDE decrease the entropy of π_c in all configurations, as we can observe from Figs. 19 and 20. In particular, SDE markedly reduces Eve's estimation accuracy, but also more substantially impairs the reward compared to PDE.

In Fig. 21, we focus on the scenario with $\beta = 1$ and $\theta = 32$ and analyze the impact of the time gap D on all the proposed strategies. Fig. 21(a) shows that MPI presents $\mathbb{E}[L_E] \approx 0.6$ for $D = 5$, only slightly higher than the one obtained with ADE. On the other hand, MPI becomes more vulnerable as D increases, while the average leakage of ADE never exceeds

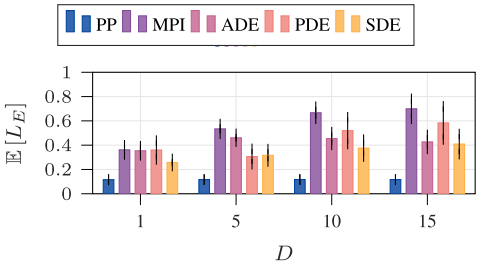


(a) Leakage. (b) Total reward.

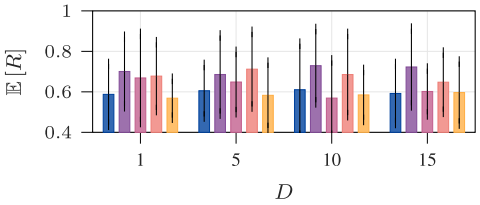


(c) Control reward. (d) Eve's accuracy.

Fig. 20. SDE performance as a function of θ and β in the control scenario, with $D = 5$.



(a) Information leakage.



(b) Total reward.

Fig. 21. Expected leakage and reward as a function of the opacity time gap D in the control scenario, with $\beta = 1$ and $\theta = 32$.

$L_{\text{high}} = 0.6$. The PDE heuristic is more robust than ADE for $D \leq 5$, while SDE is the most effective solution from the point of view of secrecy for all the D configurations. If we consider the total reward, reported in Fig. 21(b), the relationship between ADE and PDE is more complex: PDE has a degraded performance for $D \leq 5$, while it outperforms ADE for longer opacity time gaps. Lastly, we note that SDE exhibits substantially degraded performance, yielding a reward inferior to that of periodic transmission.

D. Pareto Analysis

So far, we have considered specific hyperparameters for all heuristic strategies. In the following, we study the trade-off between secrecy and reward for ADE, PDE and SDE

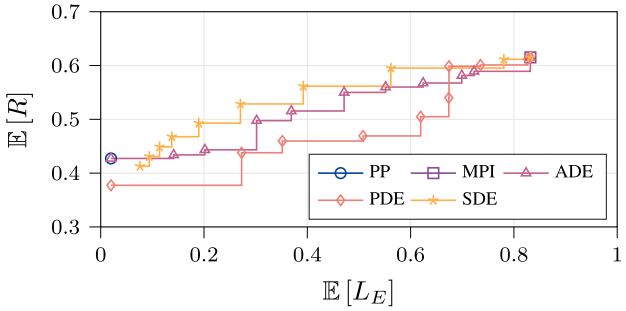


Fig. 22. Pareto frontier of the trade-off between information leakage and reward in the estimation scenario, with $\beta = 1$, $\theta = 32$, and $D = 5$.

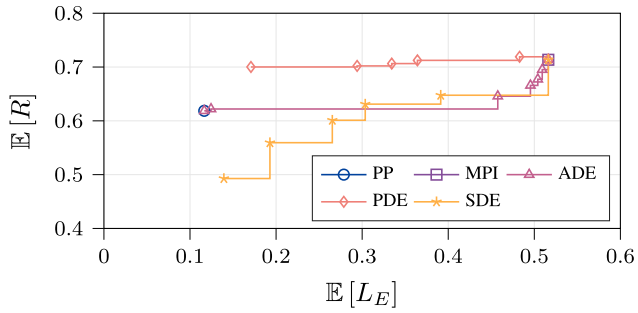


Fig. 23. Pareto frontier of the trade-off between information leakage and reward in the control scenario, with $\beta = 1$, $\theta = 32$, and $D = 5$.

by computing the Pareto frontier obtained by considering all the possible configurations for each algorithm. We vary the low leakage threshold of ADE $L_{\text{low}} \in [0.1, 0.7]$ while setting $L_{\text{high}} = L_{\text{low}} + 0.2$, the target entropy $H^* \in \left[0, H(\pi_c^{\text{MPI}})\right]$ for PDE, and the softmax temperature $\nu \in [0, 1]$ for SDE. To obtain reliable results, we run a total of $N_{\text{ep}} = 50$ independent episodes for each configuration.

Fig. 22 focuses on the remote estimation case. The marker in each plot show the achievable configurations for each algorithm, and ideal results are in the upper left corner. The results show that ADE always outperforms PDE when the leakage is lower than 0.7, while SDE outperforms them both, except for cases in which extreme secrecy is required, i.e., the target leakage is lower than 0.1. This is due to the iterative nature of PDE in identifying the defensing strategy: suboptimal choices in the early stages of the algorithm significantly degrade performance for all the following steps. This phenomenon is reflected by the steep performance drop experienced by PDE for $\mathbb{E}[L_E] \approx 0.7$, which never recovers and is always outperformed by ADE. On the other hand, SDE more effectively controls the trade-off between secrecy and efficiency compared to both ADE and PDE. The randomness in the decisions improves secrecy, while limiting the damage to the legitimate application's reward.

Fig. 23 reports the Pareto analysis for the control scenario. In this case, PDE finds a working trajectory that allows Bob to maintain a reward very close to the optimum even when reducing the leakage to values lower than 0.2. In contrast, ADE and SDE are both inefficient in managing the control system, as they immediately lead to a reward degradation as we consider $\mathbb{E}[L_E] \leq 0.5$. This may be due to the need for specific inter-transmission intervals in some crucial

states, without which control becomes much more difficult. PDE is the only strategy that can consider this and provide a deterministic outcome. The fact that the action policy is mutually adapted to communication decisions ensures that the system experiences a negligible performance loss.

PDE and SDE are both more effective than ADE, the former in the control scenario, the latter in the estimation scenario. These defenses can effectively control the trade-off between performance and secrecy: while MPI and PP provide the two extreme choices when one of the two is of paramount importance, the proposed defense schemes allow system designers to balance the two in contexts in which they both matter to different degrees.

VII. Conclusion and Future Work

In this work, we analyzed the security of pull-based Goal-oriented Communication (GoC) systems for remote control of Markov processes, focusing on the vulnerability to side channel attacks. This type of attack is viable even under information-theoretic secrecy, as they only rely on the presence of a message rather than its content. We considered two different tasks, i.e., remote estimation and remote control, and analyzed five different transmission protocols: the optimal GoC policy, a periodic transmission policy, and three heuristic solutions that trade off between the previous approaches.

Our results prove that goal-oriented scheduling has significant performance benefits, but is also highly vulnerable to eavesdropping. However, finding the optimal defense policy to balance performance and secrecy under game-theoretic rationality is a computationally hard problem, which makes it necessary to adopt heuristic mitigation strategies. We showed that any heuristic strategy must be tuned according to the target environment, as the most effective defense may vary significantly depending on factors such as the randomness of the system evolution and the communication cost.

As our study is the first to analyze timing attacks against GoC, there are many possible avenues for future work. First, expanding the game-theoretic model may lead to more efficient heuristics. It will be interesting to consider reinforcement learning solutions, which have properties similar to the proposed algorithms and can be deployed in more complex real-world applications, as well as in dynamic environments whose statistics are not stationary and to which the agents may need to adapt in real time, or deal with significant communication delays. Finally, our framework could be applied to push-based scenarios in which the transmitter independently decides when to send an update, which represents another attractive possibility for future research.

Appendix
Extension to Differential Privacy

Throughout this work, we assumed that Alice's transmissions are perfectly secure, in order to highlight the threat posed by timing attacks. However, modern semantic and GoC systems cannot offer perfect secrecy, but only differential privacy guarantees [9], [10]. This means that Eve can glean some information about the state of the system from the content of packets [11], but her knowledge is strictly limited. Recent developments have also considered systems that can operate at different points in the trade-off between privacy and communication performance [45]. This Appendix presents an extension of the model to differential privacy, showing its adaptability to practical use cases.

In the differential privacy scenario, Alice's k-th transmission is associated with an observation $o_k \in \mathcal{O}$ for Eve. The probability $\Omega_{s,o}$ of observing o when Alice transmits state s is known to all participants, and these probabilities are collected in matrix $\mathbf{\Omega} \in [0,1]^{|\mathcal{S}|\times|\mathcal{O}|}$. Differential privacy requirements are often imposed as lower bounds on $H(S|O)$, which can be easily computed from $\mathbf{\Omega}$. However, as we saw in the perfect secrecy case, timing attacks can provide further information about the state of the underlying Markov process.

In the following, we extend the Hidden Markov Model (HMM) presented in Sec. IV to consider the combination of timing-based and traditional eavesdropping. After the k-th transmission, Eve receives a combined observation $\langle \tau(k), o(k)\rangle$, where the latter depends on the transmitted state $s(k)$, while the former depends on Bob's decision and is based on the previous transmitted state $s(k-1)$. Therefore, Eve must then track $s(k)$ and $s(k-1)$ together to maintain the Markov property of the hidden state. We then define a new HMM whose hidden state space is $\mathcal{H} = \mathcal{S} \times \mathcal{S}$, with $h(k) = \langle s(k-1), s(k)\rangle$. Hence, we can easily define the observation probability as $p(\langle \tau, o\rangle | \langle s, s'\rangle) = \sigma(\tau|s)\Omega_{s',o}$. The transition probability matrix of the model, $\mathbf{F} \in [0,1]^{|\mathcal{S}|^2\times|\mathcal{S}|^2}$, is simply defined by $F_{\langle s,s'\rangle,\langle s'',s'''\rangle} = \zeta_{\sigma(s'),s'}(s''')\delta(s',s'')$. In other words, the first element of the hidden state at step $k+1$ is $s(k)$ and must be identical to the second element of the hidden state at step k for the transition to be possible.

We can then update the forward pass in (8) as

$$f_k(s,s') = \sum_{s''\in\mathcal{S}} F_{\langle s'',s\rangle,\langle s,s'\rangle}\sigma(\tau|s)\Omega_{s',o(k)} f_{k-1}(s'',s). \quad (20)$$

Similarly, the backward pass in (9) is updated as

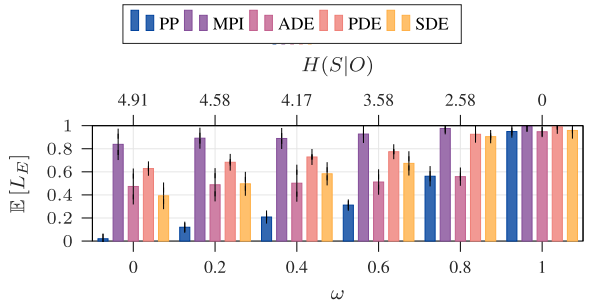
$$b_k(s,s';n) = \delta(\tau(k+1),\sigma(s))\Omega_{s',o(k)} \times \sum_{s''\in\mathcal{S}} F_{\langle s,s'\rangle,\langle s',s''\rangle} b_{k+1}(s',s'';n). \quad (21)$$

The MAP estimate of the state from (10) then becomes

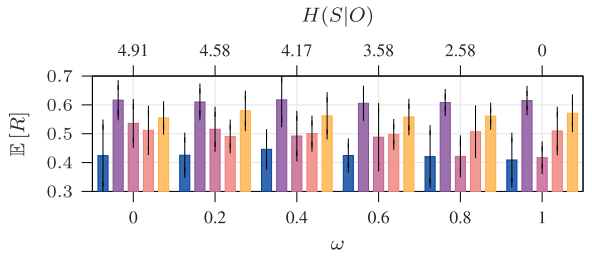
$$\phi_k(s;n) = \frac{\sum_{s''\in\mathcal{S}} f_k(s'',s) b_k(s'',s;n)}{\sum_{s'',s'''\in\mathcal{S}} f_k(s'',s''') b_k(s'',s''';n)}. \quad (22)$$

Finally, the intermediate step estimate in (11) remains unchanged, using the modified definition of $\phi_k(s;n)$. This modified HMM can be used by Eve to combine the timing signal with the observation associated with each transmission, obtaining a more accurate estimate of the state. The running time of the forward-backward algorithm becomes $O(|\mathcal{S}|^3 n)$, but is still manageable for small Markov processes.

To evaluate the effects of differential privacy on the model performance, we consider a system where the cardinality of the observation space of Eve is identical to the number of states, i.e., $|\mathcal{S}| = |\mathcal{O}|$, and denote by i_s and i_o the indexes of state s and observation o, numbered from 1 to 30.

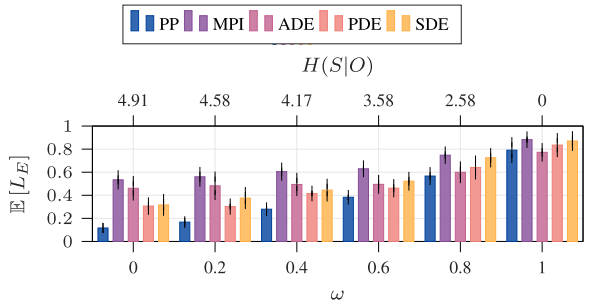


(a) Information leakage.

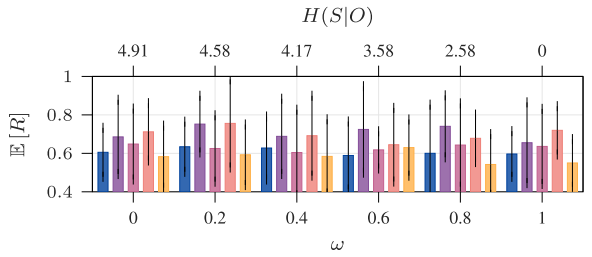


(b) Total reward.

Fig. 24. Expected leakage and reward as a function of the reconstruction capability ω in the estimation scenario, with $\beta = 1$, $\theta = 32$ and $D = 5$.



(a) Information leakage.



(b) Total reward.

Fig. 25. Expected leakage and reward as a function of the reconstruction capability ω in the control scenario, with $\beta = 1$, $\theta = 32$ and $D = 5$.

We introduce a *message secrecy* parameter $\omega \in [0, 1]$, such that the observation probability PMF $\mathbf{\Omega}_s$ when Alice transmits state s, i.e., the s-th row of matrix $\mathbf{\Omega}$, corresponds to a uniform distribution in $\left\{s + \frac{2-\omega}{2|\mathcal{S}|}, \ldots, s - \frac{2-\omega}{2|\mathcal{S}|}\right\}$. If $\omega = 1$, messages are perfectly decipherable by Eve, as the distribution collapses into a deterministic state observation. On the other hand, if $\omega = 0$, the observation follows a uniform distribution over the whole state space, corresponding to perfect message secrecy. In this case, Eve receives no additional information except for the inter-transmission intervals τ, which corresponds to the scenario studied in the main body of the paper.

In Fig. 24 and Fig. 25, we analyze the performance of MPI, PP, and the heuristic strategies in the estimation and control tasks, respectively, fixing the settings to $\beta = 1$, $\theta = 32$, and $D = 5$ and varying the value of $\omega \in [0, 1]$. As expected, as $\omega \to 1$, the expected leakage increases for all the strategies. This is especially evident for PP, which ensures perfect secrecy for $\omega = 0$ and leaks the same information as the optimal GoC strategy for $\omega = 1$. Interestingly, the leakage seems to increase approximately linearly with ω.

The expected reward $\mathbb{E}[R]$ does not vary for any policy but ADE, which tries to apply some countermeasures against the adversarial attack. As $\omega \to 1$, ADE switches to PP more frequently: this, on one hand, ensures that $\mathbb{E}[L_E]$ never exceeds L_{high} for $\omega < 0.8$, while, on the other hand, it leads to a reward similar to the periodic policy for $\omega \geq 0.8$. Incorporating differential privacy makes it essential to actively quantify leakage, thereby favoring online algorithms such as ADE. Modifications to PDE and SDE to make them aware of message leakage are an interesting avenue for future work.

REFERENCES

[1] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL, USA: Univ. of Illinois Press, Sep. 1949.

[2] D. Gündüz et al., "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 5–41, Jan. 2023.

[3] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sep. 2019.

[4] E. Fountoulakis, N. Pappas, and M. Kountouris, "Goal-oriented policies for cost of actuation error minimization in wireless autonomous systems," *IEEE Commun. Lett.*, vol. 27, no. 9, pp. 2323–2327, Sep. 2023.

[5] T. M. Getu, G. Kaddoum, and M. Bennis, "A survey on goal-oriented semantic communication: Techniques, challenges, and future directions," *IEEE Access*, vol. 12, pp. 51223–51274, 2024.

[6] Z. Lu et al., "Semantics-empowered communications: A tutorial-cum-survey," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 1, pp. 41–79, 1st Quart., 2024.

[7] S. Guo et al., "A survey on semantic communication networks: Architecture, security, and privacy," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 5, pp. 2860–2894, Oct. 2025.

[8] T.-Y. Tung and D. Gündüz, "Deep joint source-channel and encryption coding: Secure semantic communications," in *Proc. IEEE Int. Conf. Commun.*, May 2023, pp. 5620–5625.

[9] X. Liu et al., "SemProtector: A unified framework for semantic protection in deep learning-based semantic communication systems," *IEEE Commun. Mag.*, vol. 61, no. 11, pp. 56–62, Nov. 2023.

[10] S. Y. Kung, "A compressive privacy approach to generalized information bottleneck and privacy funnel problems," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1846–1872, Mar. 2018.

[11] W. Chen, S. Shao, Q. Yang, Z. Zhang, and P. Zhang, "A superposition code-based semantic communication approach with quantifiable and controllable security," *IEEE Trans. Mobile Comput.*, vol. 25, no. 2, pp. 2444–2461, Feb. 2026.

[12] A. Mostaani, T. X. Vu, S. K. Sharma, V.-D. Nguyen, Q. Liao, and S. Chatzinotas, "Task-oriented communication design in cyber-physical systems: A survey on theory and applications," *IEEE Access*, vol. 10, pp. 133842–133868, 2022.

[13] C. Feng, K. Zheng, Y. Wang, K. Huang, and Q. Chen, "Goal-oriented wireless communication resource allocation for cyber-physical systems," *IEEE Trans. Wireless Commun.*, vol. 23, no. 11, pp. 15768–15783, Nov. 2024.

[14] P. Talli et al., "Pragmatic communication for remote control of finite-state Markov processes," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 7, pp. 2589–2603, Jul. 2025.

[15] T. Van Goethem, W. Joosen, and N. Nikiforakis, "The clock is still ticking: Timing attacks in the modern web," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1382–1393.

[16] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 746–789, 1st Quart., 2020.
[17] F. Mason, F. Chiariotti, P. Talli, and A. Zanella, "Eavesdropping on goal-oriented communication: Timing attacks and countermeasures," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2025, pp. 1–6.
[18] J. Xu, B. Ai, W. Chen, N. Wang, and M. Rodrigues, "Deep joint source-channel coding for image transmission with visual protection," *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 6, pp. 1399–1411, Dec. 2023.
[19] Y. Li, Z. Shi, H. Hu, Y. Fu, H. Wang, and H. Lei, "Secure semantic communications: From perspective of physical layer security," *IEEE Commun. Lett.*, vol. 28, no. 10, pp. 2243–2247, Oct. 2024.
[20] J. Shi, Q. Zhang, W. Zeng, S. Li, and Z. Qin, "Secure transmission in wireless semantic communications with adversarial training," *IEEE Commun. Lett.*, vol. 29, no. 3, pp. 487–491, Mar. 2025.
[21] Y. Rong et al., "Semantic entropy can simultaneously benefit transmission efficiency and channel security of wireless semantic communications," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 2067–2082, 2025.
[22] X. Huang, L. Zeng, Y. Lu, and J. An, "Secure and robust joint source-channel coding with semantic clustering and adversarial purification," *IEEE Trans. Cognit. Commun. Netw.*, vol. 12, pp. 204–216, Jan. 2026.
[23] S. Tang, Y. Chen, Q. Yang, R. Zhang, D. Niyato, and Z. Shi, "Towards secure semantic communications in the presence of intelligent eavesdroppers," 2025, *arXiv:2503.23103*.
[24] X. Xu et al., "CSBA: Covert semantic backdoor attack against intelligent connected vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 11, pp. 17923–17928, Nov. 2024.
[25] M. Shen et al., "Secure semantic communications: Challenges, approaches, and opportunities," *IEEE Netw.*, vol. 38, no. 4, pp. 197–206, Jul. 2024.
[26] Z. Yang, M. Chen, G. Li, Y. Yang, and Z. Zhang, "Secure semantic communications: Fundamentals and challenges," *IEEE Netw.*, vol. 38, no. 6, pp. 513–520, Nov. 2024.
[27] R. Meng et al., "A survey of secure semantic communications," *J. Netw. Comput. Appl.*, vol. 239, Jul. 2025, Art. no. 104181.
[28] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam, "Mitigating timing side channel in shared schedulers," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1562–1573, Jun. 2016.
[29] M.-K. Yoon, S. Mohan, C.-Y. Chen, and L. Sha, "TaskShuffler: A schedule randomization protocol for obfuscation against timing inference attacks in real-time systems," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Apr. 2016, pp. 1–12.
[30] A. Mohaisen, H. Mekky, X. Zhang, H. Xie, and Y. Kim, "Timing attacks on access privacy in information centric networks and countermeasures," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 6, pp. 675–687, Nov. 2015.
[31] L. Mazaré, "Decidability of opacity with non-atomic keys," in *Proc. 18th World Comput. Congr. (WCC)*, Aug. 2004, pp. 71–84.
[32] X. Yin and S. Lafortune, "A new approach for the verification of infinite-step and k-step opacity using two-way observers," *Automatica*, vol. 80, pp. 162–171, Jun. 2017.
[33] A. Saboori and C. N. Hadjicostis, "Verification of K-step opacity and analysis of its complexity," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 3, pp. 549–559, Jul. 2011.
[34] J. Chen, M. Ibrahim, and R. Kumar, "Quantification of secrecy in partially observed stochastic discrete event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 185–195, Jan. 2017.
[35] S. Liu, X. Yin, and M. Zamani, "On a notion of approximate opacity for discrete-time stochastic control systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 5413–5418.
[36] B. Bérard, K. Chatterjee, and N. Sznajder, "Probabilistic opacity for Markov decision processes," *Inf. Process. Lett.*, vol. 115, no. 1, pp. 52–59, Jan. 2015.
[37] Y. Ji, Y.-C. Wu, and S. Lafortune, "Enforcement of opacity by public and private insertion functions," *Automatica*, vol. 93, pp. 369–378, Jul. 2018.
[38] H. Nam, S. L. Fleming, and E. Brunskill, "Reinforcement learning with state observation costs in action-contingent noiselessly observable Markov decision processes," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, Dec. 2021, pp. 15650–15666.
[39] V. Krishnamurthy, *Partially Observed Markov Decision Processes*. Cambridge, U.K.: Cambridge Univ. Press, Mar. 2016.
[40] D. Bertsekas, *Rollout, Policy Iteration, and Distributed Reinforcement Learning*. Belmont, MA, USA: Athena Scientific, Aug. 2021.
[41] K. Horák, B. Bošanský, V. Kovařík, and C. Kiekintveld, "Solving zero-sum one-sided partially observable stochastic games," *Artif. Intell.*, vol. 316, Mar. 2023, Art. no. 103838.
[42] A. Delage, O. Buffet, J. S. Dibangoye, and A. Saffidine, "HSVI can solve zero-sum partially observable stochastic games," *Dyn. Games Appl.*, vol. 14, no. 4, pp. 751–805, Sep. 2023.
[43] G. B. Dantzig, "A proof of the equivalence of the programming problem and the game problem," in *Activity Analysis of Production and Allocation*. Hoboken, NJ, USA: Wiley, Jan. 1951, ch. 2, pp. 330–338.
[44] N. Metropolis and S. M. Ulam, "The Monte Carlo method," *J. Amer. Stat. Assoc.*, vol. 44, no. 247, pp. 335–341, 1949.
[45] W. Chen, Q. Yang, S. Shao, S. Tang, Z. Shi, and S. Yu, "Privacy-preserving semantic communication over wiretap channels with learnable differential privacy," 2025, *arXiv:2510.23274*.

Federico Mason (Member, IEEE) received the master's and Ph.D. degrees from the University of Padua, Italy, in 2018 and 2023, respectively. From 2022 to 2023, he was with the IRCCS Institute of Neurological Sciences of Bologna, Italy. He is currently an Assistant Professor with the Department of Information Engineering, University of Padua. His main research interests include the development of new techniques to process electrophysiological signals and the design of multi-agent learning strategies for network optimization.

Federico Chiariotti (Senior Member, IEEE) received the Ph.D. degree from the Department of Information Engineering, University of Padua, Italy, in 2019. From 2020 to 2022, he worked at Aalborg University, Denmark. He is currently an Assistant Professor with the Department of Information Engineering, University of Padua. His research focuses on machine learning in networks and goal-oriented communication. He has published more than 100 peer-reviewed articles and received four best paper awards. He is an Associate Editor of IEEE NETWORKING LETTERS and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Pietro Talli (Graduate Student Member, IEEE) received the bachelor's degree in information engineering in 2020 and the master's degree in ICT for internet and multimedia in 2022. He is currently pursuing the Ph.D. degree, working on semantic and goal-oriented communications. In 2024, he spent a period as a Visiting Ph.D. Student at Aalborg University, Denmark. He is currently pursuing the Ph.D. degree with the Department of Information Engineering, University of Padua, Italy. His research interests are in machine learning for communications and multi-agent systems.

Andrea Zanella (Senior Member, IEEE) received the Laurea and Ph.D. degrees in computer engineering from the University of Padua, Italy, in 1998 and 2001, respectively. He is currently a Full Professor with the Department of Information Engineering, University of Padua. He is one of the coordinators of the SIGnals and NETworking (SIGNET) Research Laboratory. His long-established research activities are in the fields of protocol design, optimization, and performance evaluation of wired and wireless networks.

Open Access funding provided by 'Università degli Studi di Padova' within the CRUI CARE Agreement