This manuscript has been submitted to IEEE Journal on Selected Areas in Communication.

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 plans data transmission to occur 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 obtain information about 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 two practical 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 goal-oriented 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 Goal-oriented Communication (GoC) paradigm has attracted a significant amount of interest from the research community. The concept was advanced by Warren 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. On the other hand, a 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 successively been extended to scheduling strategies that consider contextual and past information [4]. These initial studies have shown that GoC leads to impressive performance advantages, fostering research on more practical aspects including security against eavesdropping attacks [5]. The most common approach to enhance GoC security is to train the transmitter to encrypt the data [6], modifying the encoding mechanism to trigger an incorrect semantic interpretation by possible eavesdroppers [7], while allowing the intended receiver to decode the original message. In this regard, information-theoretic approaches [8] can provide more solid confidentiality guarantees [9], but only under specific assumptions on the nature of the encoder and decoder.

Although the above mechanisms address the risk of leaking information through the *content* of the transmitted data, another specific vulnerability of GoC systems has been mostly neglected so far: side-channel attacks that aim to infer the state of the system from the *timing* of messages [10]. This is particularly critical for Internet of Things (IoT) applications or other resource-constrained monitoring systems, where GoC is used to reduce the frequency of updates according to the status of the monitored process. In these scenarios, timing attacks can leak information about the content of transmitted packets (i.e., the state of the process) even when using one-time pad encryption or information-theoretic security.

In this work, we analyze the secrecy of a goal-oriented scheduling system under a timing attack from an eavesdropper. Specifically, we consider a pull-based communication scenario in which a controller node maintains an online estimate of the state of a remote Markov process, in order to monitor or control the process itself [11]. The state of the process is not directly observable by the controller node but is continuously tracked by a sensor node that can transmit the current state to the other node upon request. The goal of the controller is to schedule status update transmissions from the sensor node to obtain high reward for its local task, while minimizing the channel occupancy and limiting the information that can be inferred by an eavesdropper from the timing between consecutive transmissions in either direction.

We consider the critical condition in which the eavesdropper knows the state-transition probability matrix of the monitored Markov process and the policy used by the controller to schedule transmissions from the sensor node. In addition, the eavesdropper knows the timing of all past transmissions. Hence, we analyze the trade-off between secrecy, which depends on the information leakage of the system, and performance, measured in terms of reward for the controller node and transmission efficiency. We consider four different strategies: a pure goal-oriented approach, which optimizes performance disregarding security aspects; a periodic scheduling that prevents timing attacks, but loses GoC advantages; and two novel heuristics that reduce information leakage while preserving performance. The analysis is repeated both in a monitoring scenario, where

The authors (emails: federico.mason@unipd.it, federico.chiariotti@unipd.it, pietro.talli@phd.unipd.it, andrea.zanella@unipd.it) are with the Department of Information Engineering, University of Padova, Via G. Gradenigo 6/B, 35131, Padova, Italy. This project was funded under the National Recovery and Resilience Plan (NRRP), funded by the European Union NextGenerationEU Project as part of the "RESTART" partnership (PE0000001).

the controller only aims at estimating the status of the process, and in a control scenario, where the system's evolution can be altered by the controller itself.

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.
- We propose a heuristic algorithm, named Alternating Defense from Eavesdropping (ADE), which allows the legitimate agent to compute the information leakage in real time and take countermeasures accordingly.
- We propose a lighter heuristic algorithm, named Packing Defense from Eavesdropping (PDE), which pursues the same objective as ADE, but with a lower complexity, enabling its implementation as a look-up table.
- We evaluated the effectiveness of timing attacks and defensive strategies through Monte Carlo simulations for both estimation and control scenarios.

A preliminary version of this work was presented as a conference paper in [12]. This manuscript extends our previous results by introducing the PDE policy and analyzing the overall framework in the case of control applications.

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 [11], 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 which focuses 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.

Besides an early work using an information bottleneck approach [8], previous studies mainly deal with eavesdropping attacks using deep learning [5]. More recently, the authors of [9] provide a near-information-theoretic security approach for semantic communication. The authors adopt the classic approach in information-theoretic security by considering 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 [6], 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 function, exploiting similar principles as traditional information-theoretic security. Interestingly, the JSCC protection module can be implemented after semantic encoding ([6]), before encoding ([13]), or integrated within the encoder ([14], [15]), with similar results and trade-offs in terms of secrecy and image transfer quality.

Another example of semantic encryption is given in [7], where eavesdroppers adopt a model inversion approach to retrieve the original information. The use of explicit semantic features of the image [16] 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 [17]. Finally, the authors of [18] 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 [19], 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 [20], [21], 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 [22]. 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, because this type of attack can be effective even when the content of messages is perfectly secure (e.g., when protected through one-time padding).

Interestingly, side-channel attacks have been considered in other practical scenarios, such as cloud scheduling. For example, the work in [23] analyzes a model in which a scheduler dispatches 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 [24], which significantly reduces information 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 user requests and the popularity of content [25].

Finally, we consider related work from another field, namely, remote estimation and control: 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 case of a 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

TABLE I
MODEL NOTATION.

Symbol	Description	Symbol	Description	Symbol	Description	Symbol	Description
$\mathcal{S}$	State space	$\mathcal{A}$	Action Space	$\boldsymbol{P}$	Transition probability matrix	γ	Discount factor
$R(\cdot)$	Total reward function	$r_B(\cdot)$	Bob's reward function	$r_A(\cdot)$	Alice's reward function	T_{max}	Maximum timing signal
β	Transmission cost	$\boldsymbol{\mu}$	Steady-state distribution	$\psi(\cdot)$	Communication policy	$\pi(\cdot)$	Control policy
D	Opacity time gap	$L_E(n;D)$	Information leakage	$\boldsymbol{\phi}_E$	Eve's belief distribution	L_{min}	Minimum leakage
η	Eve's estimate	$\sigma(\cdot)$	Communication policy	$f_k(s)$	Forward probability	$b_k(s;n)$	Backward probability
L_{low}	ADE's lower threshold	L_{high}	ADE's higher threshold	$\xi_\sigma^{(s^*,\tau)}(\cdot)$	Single deviation policy	H^*	PDE's target entropy
θ	Density decay	$H(\cdot)$	Entropy function	$\zeta_{\tau,s}(s')$	τ-step transition probability	$\delta(\cdot)$	Kronecker delta function

opaque if an eavesdropper with limited observations is unable to estimate some restricted information [26], 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 [27] and even scenarios in which the eavesdropper has access to the entire observation history [28]. In information-theoretic terms, opacity can be defined as the difference between the entropy of the belief distribution of the legitimate monitor and that of the eavesdropper [29].

In control scenarios, where the legitimate agent can affect the state evolution of the system through actions, but the control policy is known to the eavesdropper, opacity is more difficult to achieve, and its formal verification becomes a highly complex [30] or even undecidable problem [31]. At the same time, the ability to affect the state of the system enables agents to actively improve security by inserting fictitious events [32] to confuse eavesdroppers. This inherent complexity makes it critical to design GoC policies that optimize control performance under opacity constraints, or optimize both simultaneously. To the best of our knowledge, the current literature considers only the problem of maximizing the opacity of the initial state or the current state. In this work, we generalize the problem considering the opacity of the entire system history, which is a significantly more challenging problem.

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 $\mathbf{P}$. 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 a state space $\mathcal{A}$. Both Alice and Bob have complete knowledge of $\mathbf{P}$ 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. The notation used in our model is reported in Tab. I.

We consider a *pull-based* configuration in which, at each time step n, Bob must decide whether to ask Alice for an update, thus incurring a communication cost $\beta \in \mathbb{R}^+$, or to estimate the current state of the Markov chain from the information he already knows. 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. This parameter is necessary for the tractability of the analysis, but its impact can be arbitrarily minimized by considering large values of $T_{\max}$.

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 function is equal to 1 if the state estimate matches the actual state, and 0 otherwise, i.e., $r_B(s,\hat{s}) = \delta(s,\hat{s})$, where $\delta(\cdot,\cdot)$ is the Kronecker delta function.

We also introduce the *communication reward function* $r_A : \{0,1\} \to \mathbb{R}$, with $r_A(c) = -\beta c$, where β is a communication 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:

$$R(s,c,a) = r_B(s,a) + r_A(c). \tag{1}$$

Therefore, Bob's objective is to find the communication policy that maximizes the expected cumulative reward

$$G(n) = E\left[\sum_{k=n}^{+\infty} \gamma^{(k-n)} R(s(k),c(k),a(k))\right], \tag{2}$$

where $\gamma \in [0,1)$ is the exponential discount factor. The described problem is a remote Partially Observable Markov Decision Process (POMDP), comprehensively characterized by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{P}, r_B(\cdot), \gamma, T_{\max}, \beta \rangle$.

We assume that the communication delay is shorter than the time step of the underlying Markov process, so that when Alice transmits, Bob receives the state information instantaneously (i.e., 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 to say, a *belief* on the process state that we denote as $\boldsymbol{\zeta}$.

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

$$\zeta_{\Delta,s}(s') = \sum_{s'' \in \mathcal{S}} P(s'|s'';\pi)\zeta_{\Delta-1,s}(s''), \tag{3}$$

with $\zeta_{0,s}(s') = \delta(s,s')$. Bob's control policy π is a parameter of the transition probabilities because, in the control scenario, the evolution of the Markov process is generally affected by Bob's actions. In the estimation case, we can simplify (3) to

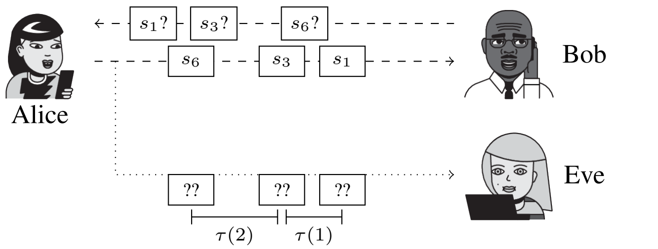


Fig. 1. The goal-oriented eavesdropping attack: Eve cannot decipher Alice's updates, but the timing signal τ allows her to estimate the state of the remote Markov process.

$\zeta_{\Delta,s}(s') = \mathbf{P}^{\Delta}(s,s')$, i.e., to the element with indices s and s' of the Δ-th power of the transition matrix, as the evolution of the system does not depend on the control policy π.

Since each transmission represents a renewal of Bob's beliefs, the current estimate of the process state can be summarized by the last received state s and the time Δ since the last update [11]. Therefore, Bob's optimal decisions depend only on (s, Δ), which reduces the complexity of the problem. Importantly, the Modified Policy Iteration (MPI) scheme given in [11, Alg. 1] can find the jointly optimal GoC policy $\psi : \mathcal{S} \times \mathbb{Z}^+ \to \{0,1\}$ and the associated control policy $\pi : \mathcal{S} \times \mathbb{Z}^+ \to \mathcal{A}$, in polynomial time with respect to the size $|\mathcal{S}|$ of the state space.

Practically, any time Bob receives a state update $s \in S$ from Alice, he can determine his future control actions in advance, as well as the optimal number of time steps to wait before the next update request, which is the smallest Δ such that $\psi(s,\Delta) = 1$. We denote the transmission request scheduling function as $\sigma : \mathcal{S} \to \mathbb{Z}^+$, defined as

$$\sigma(s) = \inf\{\Delta \in \mathbb{N} : \psi(s,\Delta) = 1\}\,. \tag{4}$$

This function then determines the inter-transmission intervals.

IV. Eavesdropping Attack

We assume that an eavesdropper (Eve) knows the Markov process statistics represented by $\mathbf{P}$ and $\boldsymbol{\mu}_0$, and the transmission request scheduling $\sigma(\cdot)$. 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. A scheme of the overall scenario is reported in Fig. 1.

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 $H(\mathbf{p}) = -\sum_{s\in\mathcal{S}} p(s)\log_2(p(s))$, with $p(\cdot)$ denoting the probability distribution 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 $\boldsymbol{\mu}$ of the system. This implies that the leakage can never be less than

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

Therefore, zero leakage can be achieved only for processes with a uniform steady-state distribution $\boldsymbol{\mu}$, for which $H(\boldsymbol{\mu}) = H_0$, whereas $L_{\text{min}} > 0$ in the general case. Finally, we observe that Eve's best estimate of state $s(n)$ is obtained at step $n+D$, as she has additional observations to draw on. Accordingly, the accuracy of this estimate is

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

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

B. Forward-Backward State Estimation

Since Eve sees the system as an HMM, the MAP estimate of the process state can be computed through the forward-backward algorithm. Practically, 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 from 0 to m, while the backward probabilities are based on those from $m+1$ to n.

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

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

where $\tau(k)$ is the number of steps between transmissions $k-1$ and k, $\boldsymbol{\zeta}_{\Delta,s}$, given by (3), is the state probability distribution

[1] We specify that our formulation does not consider the initial knowledge of Eve 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.

in Δ steps assuming that the initial state was s, and $\sigma(\cdot)$ is the transmission scheduling policy defined in (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

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

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 MAP estimate of 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 MAP estimate 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. We observe that the running time of the forward-backward algorithm is $O(|\mathcal{S}|^2 n)$. Therefore, it has a relatively low energy cost, which can be further reduced by limiting n to the mixing time of the Markov chain.

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 wants to estimate the state of the remote Markov process exploiting the correlation between the state transitions of the system and the timing between Alice's transmissions.

For a given opacity time gap D, the performance of the system can be defined as the expected 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 estimation accuracy. Therefore, Bob's optimal strategy should maximize (12), while Eve's best response consists of using the forward-backward algorithm to update her estimate of the Markov process.

We can model this system as a zero-sum one-sided partially observable stochastic game (OPOSG) [33]. The solution for the game is a Nash Equilibrium (NE) where any unilateral deviation from a player's policy would result in a decrease in that player's performance. Methods to solve zero-sum OPOSGs have recently been proposed, based on the convexity property of the value function [33] or on dividing the problem into sub-games with limited trajectories [34]. However, complexity grows exponentially with the state space size. In fact, we can prove the following statement 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 [35] proves that a two-player zero-sum game with payoff matrix $\mathbf{M}$ is equivalent to the following linear programming problem:

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

Normalizing $\mathbf{x}$ returns the optimal mixed strategy for one of the players. In our case, the action space for Bob is equivalent to the possible communication and control policies that he can adopt, which grows at least exponentially with the number of states $|\mathcal{S}|$. The length of $\mathbf{x}$ will then also grow exponentially with $|\mathcal{S}|$, making the game 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 two solutions to attain this objective: Alternating Defense from Eavesdropping (ADE), which alternates between goal-oriented and periodic transmission, and Packing Defense from Eavesdropping (PDE), which is designed to reduce the entropy of Bob's scheduling decisions, thus increasing the communication opacity and making the system inherently more secure.

A. Alternating Defense

We know that the optimal GoC scheduling 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 [11, 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 policy 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 scheduling policy with period T, we have $\sigma(s) = T\ \forall\, s \in \mathcal{S}$ 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 control policy π 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.

We take advantage of this principle to design our first heuristic policy, Alternating Defense from Eavesdropping (ADE),

Algorithm 1 Alternating Defense from Eavesdropping (ADE)

```
function SCHEDULE($s, \sigma, T, \mathbf{P}, \mathbf{f}, \mathbf{b}, \boldsymbol{\tau}, L_{\text{low}}, L_{\text{high}}, \xi$)
    if $\xi = 0$ then                                        ▷ GoC active
        if $L_E(\sigma(s)) \geq L_{\text{high}}$ then              ▷ Check secrecy threshold
            return next update in $T$ steps, $\xi = 1$           ▷ Switch to PP
        else
            return next update in $\sigma(s)$ steps, $\xi = 0$   ▷ Keep using GoC
    else                                                     ▷ PP active
        if $L_E(T) < L_{\text{low}}$ then                      ▷ Check performance threshold
            return next update in $\sigma(s)$ steps, $\xi = 0$   ▷ Switch to GoC
        else
            return next update in $T$ steps, $\xi = 1$           ▷ Keep using PP
end function
```

whose pseudocode is reported as Algorithm 1. 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, Bob can switch to a Periodic Policy (PP) whenever the expectation of future leakage increases beyond an upper threshold L_{high} and switch back to GoC 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.

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 scheduling policy σ as

$$H(\sigma) = -\sum_{\tau=1}^{\infty} \frac{\sum_{s\in\mathcal{S}} \delta(\tau, \sigma(s))}{|\mathcal{S}|} \log_2 \left(\frac{\sum_{s\in\mathcal{S}} \delta(\tau, \sigma(s))}{|\mathcal{S}|} \right). \tag{14}$$

We can assume that $H(\sigma)$ 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, i.e., when $\forall\ s', s'' \in \mathcal{S},\ s' \neq s''$, we have $\sigma(s') \neq \sigma(s'')$. 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.

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

$$\xi_\sigma^{(s^*,\tau)}(s) = \tau\delta(s, s^*) + \sigma(s)(1 - \delta(s, s^*)). \tag{15}$$

Starting from the purely goal-oriented policy, denoted by $\sigma^{(0)}$, 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 $\sigma^{(i)}(s) = \xi_{\sigma^{(i-1)}}^{(s_i^*,\tau_i)}(s)$ for all $s \in S$, where

$$(s_i^*, \tau_i) = \underset{(s^*,\tau):H\left(\xi_{\sigma^{(i-1)}}^{(s^*,\tau)}\right)<H(\sigma^{(i-1)})}{\arg\max} \mathbb{E}\left[R_B | \xi_{\sigma^{(i-1)}}^{(s^*,\tau)}\right]. \tag{16}$$

Algorithm 2 Packing Defense from Eavesdropping (PDE)

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

This packing rule ensures that the new policy $\sigma^{(i)}$ is the one that maximizes the expected system reward $\mathbb{E}[R_B]$ among those with entropy lower than $H\left(\sigma^{(i-1)}\right)$. 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.

VI. Simulation Settings and Results

In the following, we study our GoC model in two 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, computed via the Modified Policy Iteration (MPI) algorithm, and the optimal 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 function $P : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ (corresponding to the matrix $\boldsymbol{P}$) depends on a single parameter θ named *density decay*, that makes it possible to tune the predictability of the evolution of the system. Specifically, we have

$$P(s,s',a)=\begin{cases}\frac{2-2g(s,\theta)}{6}, & s'=\chi(s,a)\oplus 1,\ \mathrm{mod}(s,4)=2;\\ \frac{2+g(s,\theta)}{6}, & s'=\chi(s,a)\oplus 3,\ \mathrm{mod}(s,4)=2;\\ \frac{2+g(s,\theta)}{6}, & s'=\chi(s,a)\ominus 2,\ \mathrm{mod}(s,4)=2;\\ \frac{1+2g(s,\theta)}{3}, & s'=\chi(s,a)\oplus 1,\ \mathrm{mod}(s,4)\neq 2;\\ \frac{1-g(s,\theta)}{3}, & s'=\chi(s,a)\oplus 3,\ \mathrm{mod}(s,4)\neq 2;\\ \frac{1-g(s,\theta)}{3}, & s'=\chi(s,a)\ominus 2,\ \mathrm{mod}(s,4)\neq 2;\\ 0, & \text{otherwise};\end{cases} \tag{17}$$

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

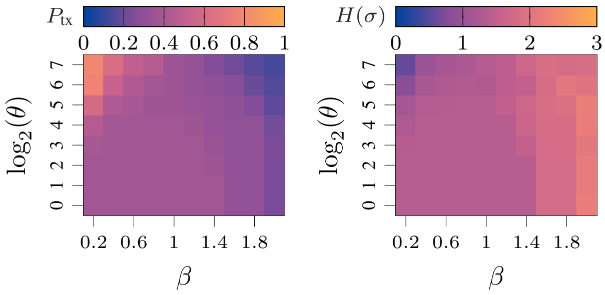
$$g(s,\theta)=\left|\frac{2(s-2)}{|\mathcal{S}|-2}-1\right|^{\theta}\in[0,1]. \tag{18}$$

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.

Hence, 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 $\chi(s,a)\oplus 1$ is more probable 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. We observe that 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, hence, fully predictable) transitions. Finally, we note that $g(s,\theta)=1$ for the extreme states $s=1$ and $s=|S|$, and progressively decreases 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, that is, 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 POMDP configurations, varying the density decay $\theta\in[1,2^7]$ and the transmission cost $\beta\in[0.2,2]$. For each configuration, we compute the optimal GoC scheduling policy given by the MPI algorithm [11], maximizing the long-term reward of the system penalized by the communication cost, as defined in (2). Hence, we compare MPI with PP, which is the best policy among those exploiting a fixed inter-transmission period, and the two heuristics presented in Sec. V. The PDE heuristic is configured by setting $H^*=\frac{1}{2}H\left(\sigma^{(0)}\right)$ as a stopping criterion, where $\sigma^{(0)}$ is the initial scheduling policy returned by the MPI algorithm. Instead, the ADE heuristic uses $L_{\text{low}}=0.4$ and $L_{\text{high}}=0.6$ as leakage thresholds. In all cases, we set $T_{\max}=10$ as the maximum interval between consecutive transmissions, i.e., the maximum value that the scheduling function $\sigma(\cdot)$ can take.



(a) Transmission probability. (b) Scheduling policy entropy.

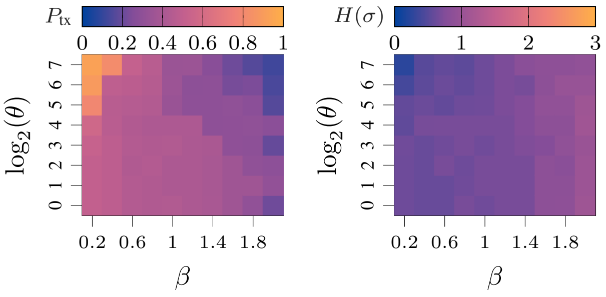
Fig. 2. Characterization of the MPI policy as a function of β and the density decay θ in the estimation scenario.

B. Remote Estimation Scenario

Focusing on the remote estimation scenario, we first analyze the characteristics of the optimal GoC policy provided by the MPI algorithm. Fig. 2a shows a heatmap of the transmission probability associated with each system configuration. We can see that 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 (which correspond to less predictable transition matrices) result in more frequent state 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 transmission probability decreases as θ increases, because the higher estimation accuracy may not cover the update cost.

Fig. 2b represents the entropy $H(\sigma)$ of the transmission scheduling policy returned by the MPI algorithm, which is a proxy of information leakage caused by transmission decisions. In general, $H(\sigma)$ decreases as $\beta\to 0$, because 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.

In general, a policy that selects a different value of $\sigma(s)$ for each state would have an entropy equal to $\log_2(|\mathcal{S}|)$, while any periodic policy would have zero entropy. Hence, we expect the MPI algorithm to have the highest entropy $H(\sigma)$ among the strategies analyzed in this paper. This is because PP uses a fixed inter-transmission period, ADE alternates between MPI and PP, while PDE is explicitly designed to reduce $H(\sigma)$ with



(a) Transmission probability.

(b) Scheduling policy entropy.

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

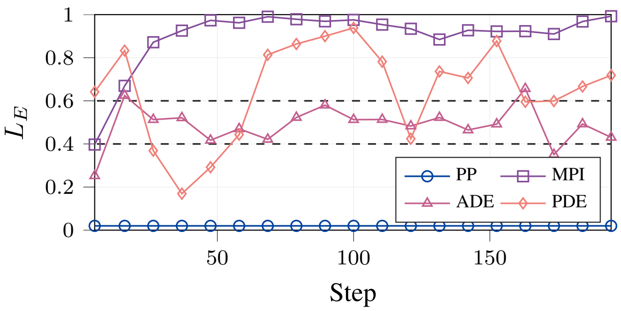
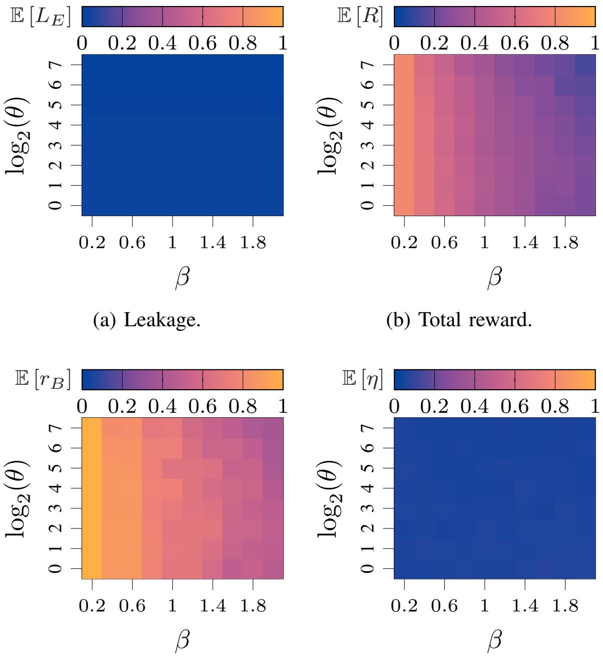


Fig. 4. Information leakage during a single episode in the estimation scenario, with $\beta = 1$, $\theta = 32$ and $D = 5$. The ADE thresholds L_{low} and L_{high} are marked as dashed lines.

respect to MPI. We can appreciate the advantages of PDE by looking at Fig. 3, where we report the transmission probability and entropy associated with the heuristic. Notably, PDE results in a transmission probability similar to MPI but successfully halves the entropy in all configurations of the system.

Although $H(\sigma)$ is a useful indicator of system opacity, the information leakage L_E, as defined in (5), provides more information on the trade-off between secrecy and performance of the remote estimation task. In Fig. 4 we then report L_E during a single episode of $N_{\text{step}} = 200$ steps, considering $\beta = 1$, $\theta = 32$, and $D = 5$. In addition to MPI and PDE, we also consider the optimal periodic strategy PP and the ADE heuristic. We can observe that the leakage of the MPI algorithm quickly approaches 1, showing that Eve correctly guesses the remote state very often with these settings. Conversely, 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}, thus offering a compromise between the two previous approaches. Finally, PDE improves secrecy compared to MPI, but the value of L_E exhibits strong oscillations over time, exposing the system to a high risk of leakage in some steps.

Fig. 5 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 $N_{\text{step}} = 200$. In addition to leakage (a), we consider the total reward R (b), defined in (1), the reward for the estimation task r_B (c), as well as the probability η that Eve correctly estimates the state of the Markov process (d), the latter defined as in (7). First, we observe that $L_E \approx 0$ for all system configurations when the PP solution is used, as expected for periodic communication. Moreover, from Fig. 5b and Fig. 5c we observe that the expected total reward $\mathbb{E}[R]$, as well



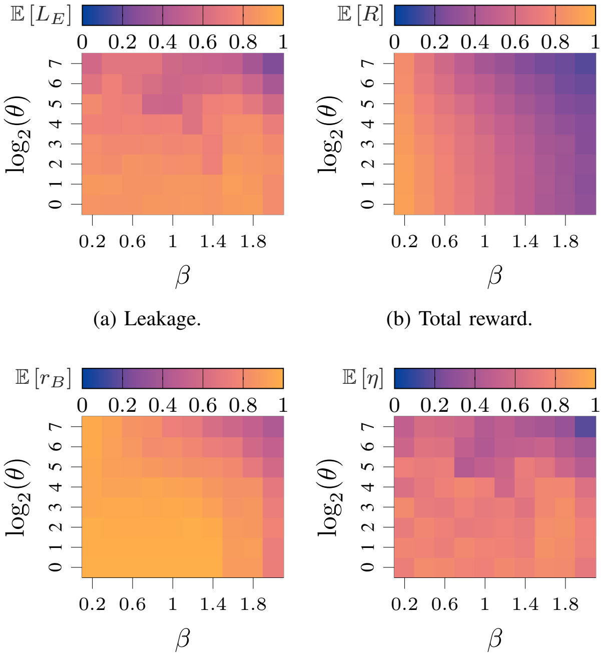
(a) Leakage.

(b) Total reward.

(c) Estimation reward.

(d) Eve's accuracy.

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



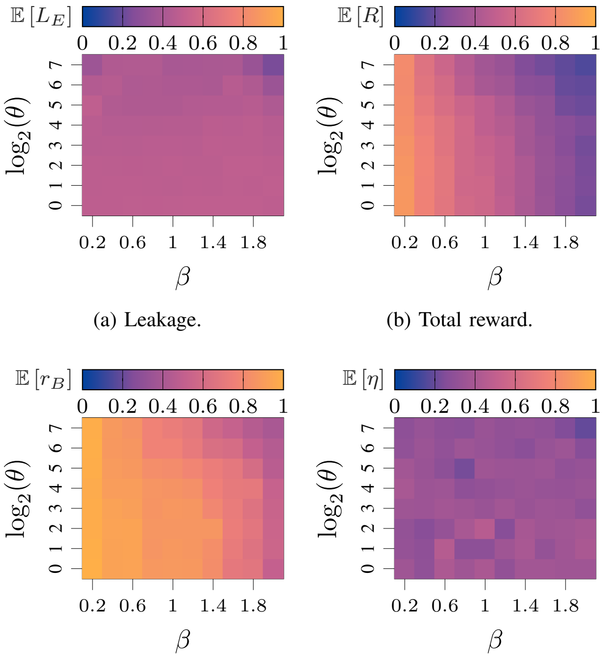
(a) Leakage.

(b) Total reward.

(c) Estimation reward.

(d) Eve's accuracy.

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



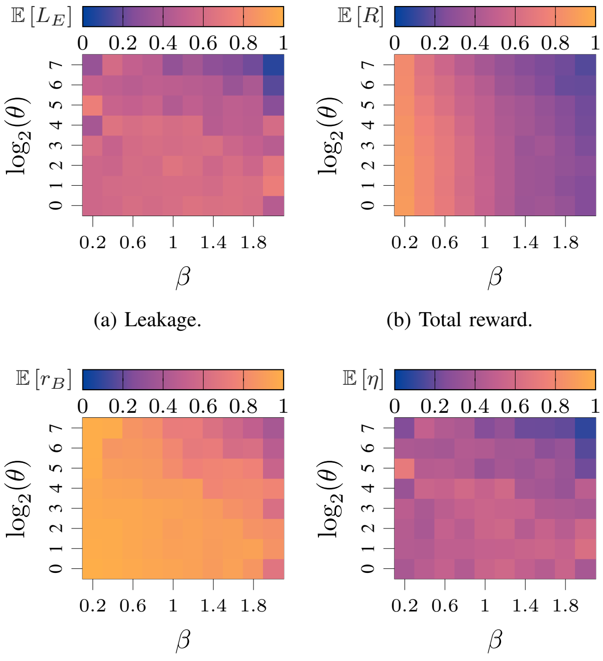
(a) Leakage.

(b) Total reward.

(c) Estimation reward.

(d) Eve's accuracy.

Fig. 7. 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.

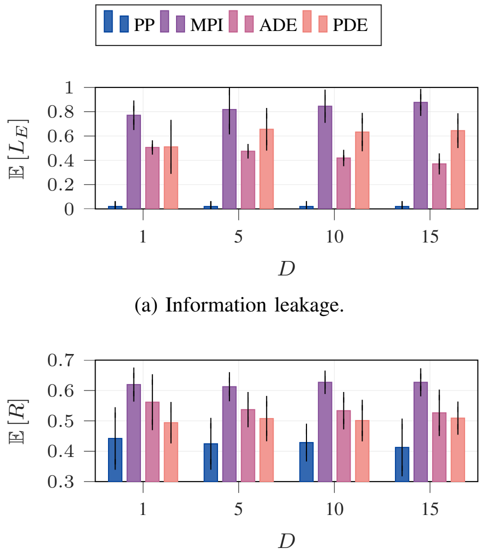
Fig. 8. PDE performance as a function of θ and β in the estimation scenario, with $D = 5$.

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. 6 offers a comparison of these performance indicators for the MPI strategy that, being purely GoC, is complementary to PP. Not surprisingly, this setting leads to a strong secrecy degradation (Fig. 6a): the information leakage is close to 0.8 for all configurations except for those with very high values of β and θ. Eve is able to correctly decode the status of the monitored process almost as often as Bob (Fig. 6d), highlighting the strong vulnerability of MPI to timing attacks. On the other hand, MPI significantly improves the total reward compared to PP (Fig. 6b). Since MPI tends to transmit more often than PP, the accuracy of Bob does not decrease significantly as β increases and the gain over PP reaches 50% when $\beta \to 2$.

The proposed heuristic strategies are expected to perform somewhere between PP and MPI. As shown in Fig. 7a, ADE improves secrecy in all configurations, guaranteeing that the leakage remains lower than L_{high}. Comparing Fig. 7c and Fig. 7d, we note that Eve's accuracy is much lower than Bob's, unlike in the MPI scenario, leading to a mean leakage of 0.45. At the same time, Fig. 7b shows that ADE degrades the total reward compared to MPI, especially in the case of Markov 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. 7b and Fig. 5b.

In Fig. 8, we report the results of PDE, which, similarly to ADE, strikes a compromise between the higher efficiency of MPI and the secrecy provided by periodic scheduling. The main difference is that PDE does not monitor information leakage explicitly, but considers the entropy of the scheduling policy as a secrecy indicator. Fig. 8a shows that the expected leakage with PDE is higher than with ADE (Fig. 8d) without significantly improving Bob's performance. Hence, ADE performs better than PDE in this remote estimation task; however, the higher computational complexity may make ADE unsuitable for implementation on nodes with limited hardware.

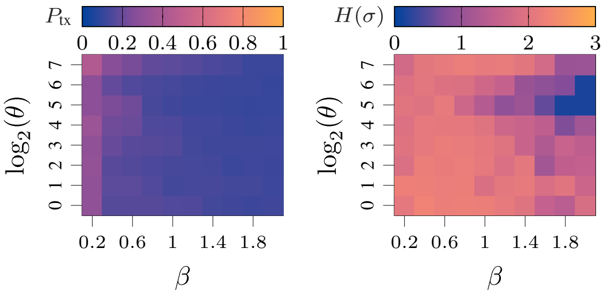


(a) Information leakage.

(b) Total reward.

Fig. 9. Expected leakage and reward as a function of D in the estimation scenario, with $\beta = 1$ and $\theta = 32$.

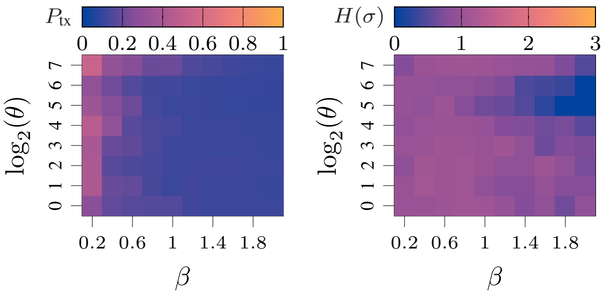
Fig. 9 analyzes the impact of the time gap D on overall performance, focusing on a system with $\beta = 1$ and $\theta = 32$, and setting $D \in \{1, 5, 10, 15\}$. As expected, the performance



(a) Transmission probability.

(b) Entropy of the scheduling policy.

Fig. 10. Characterization of the MPI policy as a function of θ and β in the control scenario.



(a) Transmission probability.

(b) Entropy of the scheduling policy.

Fig. 11. Characterization of the PDE policy as a function of θ and β in the control scenario.

of PP does not change in the different scenarios, while the leakage of MPI increases as a function of D, given that a longer time interval allows Eve to exploit more information. As the PDE scheduling policy is directly derived from MPI, PDE follows a similar trend in terms of leakage. Instead, ADE tends to make more conservative choices and switches to PP more often and for longer periods, as D increases. In particular, for $D = 15$, the reward of ADE approximates that of PDE that, for all the other configurations, shows a worse performance.

C. Remote Control Scenario

The remote control scenario has a significant difference with respect to the estimation scenario: Bob does not need to know the status of the process to maximize the reward, which depends on the closeness between the current state $s(n)$ and the target state s°. This strongly reduces the transmission probability of the MPI strategy compared to the estimation scenario. As shown in Fig. 10a, Bob updates his state estimate with high frequency only when the evolution of the process becomes highly stochastic ($\theta \gg 1$) or if the transmission cost is negligible ($\beta \to 0$). A similar trend can be observed for the PDE policy, whose transmission rate is reported in Fig. 11a and results slightly higher than that of MPI.

Fig. 10b reports the entropy of the MPI scheduling, which strongly decreases in configurations with $\theta > 16$ and $\beta > 1.4$. We can hypothesize that, 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 optimal scheduling decisions is more complex. Looking at Fig. 11b, we can appreciate how PDE follows the same pattern and, as we set $H^* = \frac{1}{2}H\left(\sigma^{(0)}\right)$, reduces the entropy of the MPI scheduling policy by 50%.

As in the estimation case, we first focus on a single episode (with $\theta = 32$, $\beta = 1$, and $D = 5$) and compare the leakage obtained by MPI, PP, and the two heuristic strategies. First, we observe that the leakage of PP is constant but has higher values than in the estimation task. This is because the steady-state distribution $\boldsymbol{\mu}(\pi)$ of the system presents higher entropy, as it is directly influenced by Bob's actions. Indeed, Bob aims to keep the current state as close as possible to s°, reducing the system's randomness, and consequently, increasing the

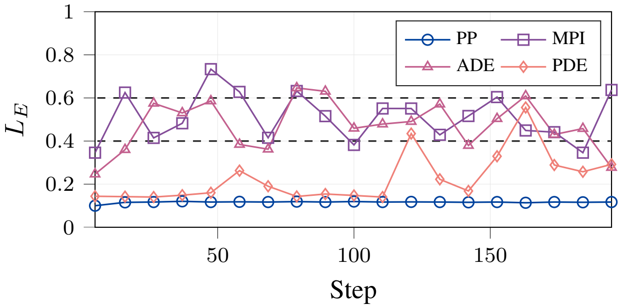
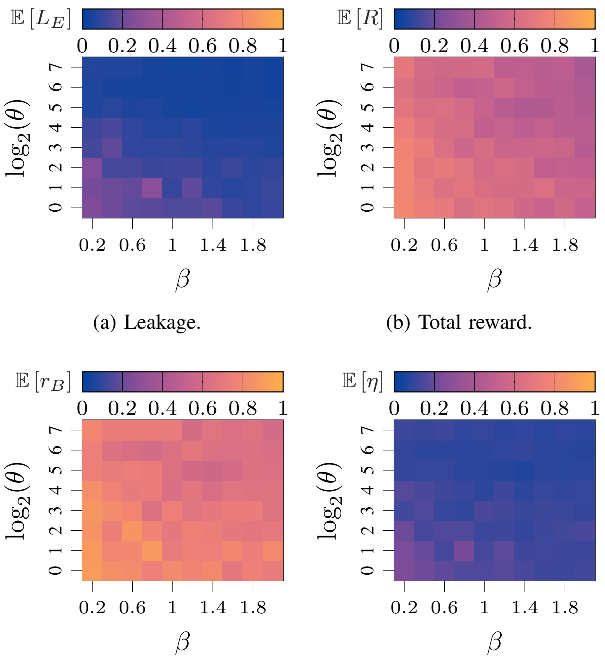


Fig. 12. Information leakage during a single episode in the control scenario, with $\beta = 1$, $\theta = 32$ and $D = 5$. The ADE thresholds L_{low} and L_{high} are marked as dashed lines.



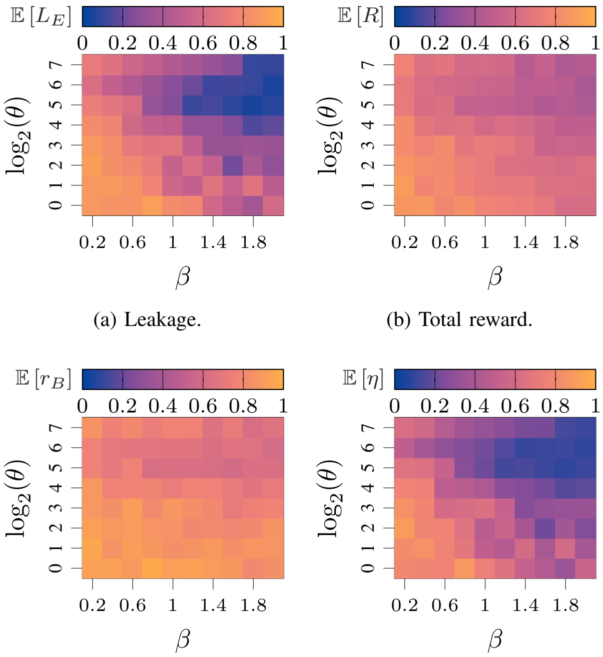
(a) Leakage.

(b) Total reward.

(c) Control reward.

(d) Eve's accuracy.

Fig. 13. PP performance as a function of θ and β in the control scenario, with $D = 5$.



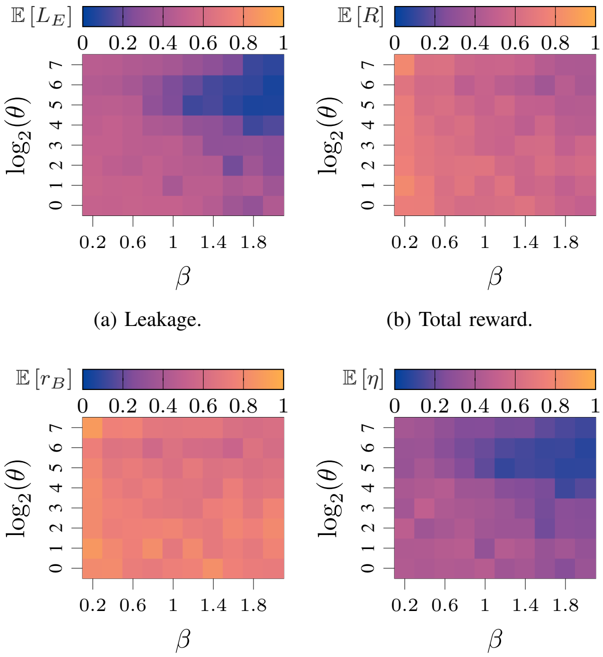
(a) Leakage.

(b) Total reward.

(c) Control reward.

(d) Eve's accuracy.

Fig. 14. MPI performance as a function of θ and β in the control scenario, with $D = 5$.



(a) Leakage.

(b) Total reward.

(c) Control reward.

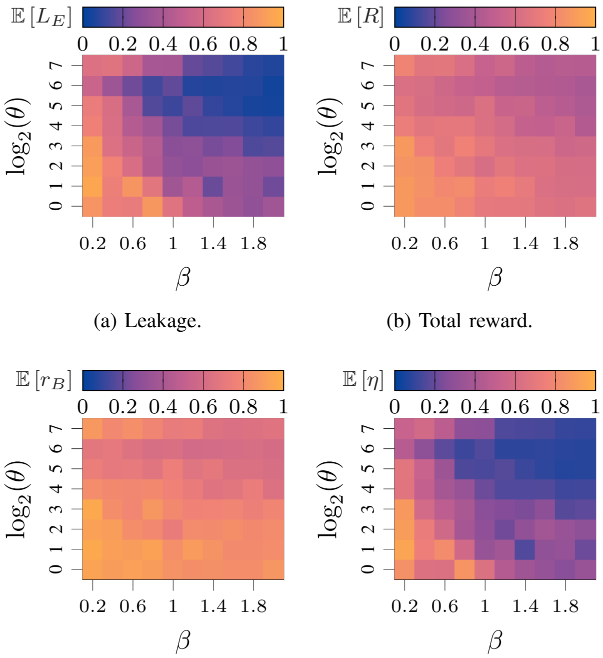
(d) Eve's accuracy.

Fig. 15. ADE performance as a function of θ and β in the control scenario, with $D = 5$.

leakage. We also note that the leakage of MPI does not increase beyond 0.7 and, consequently, ADE rarely switches to periodic communication, while PDE substantially improves secrecy with respect to both MPI and ADE.

The fact that PP may have a non-zero leakage in control tasks is confirmed by Fig. 13a, which reports the expected leakage $\mathbb{E}[L_E]$ for all combinations of density decay θ and communication cost β. Interestingly, the system is more vulnerable to timing attacks for $\beta \to 0$ and $\theta \to 0$, representing the case in which Markov transitions are more deterministic. The same configuration leads to an increase in the average reward $\mathbb{E}[r_B]$ of the control task and a slight increase of Eve's accuracy $\mathbb{E}[\eta]$ (Fig. 13c-d).

Fig. 14 reports the same analysis for the MPI approach. Comparing Fig. 14a and Fig. 10b, we observe that the information leakage strongly decreases in the region associated with a low entropy for MPI. Looking at Fig. 14d, we see that this phenomenon also affects Eve's accuracy and makes communication almost fully secret for $\beta \to 2$ and $\theta \to 2^7$. Interestingly, improvement in secrecy leads to a reduction in task reward, shown in Fig. 14d, but in a less significant manner than in the estimation scenario.



(a) Leakage.

(b) Total reward.

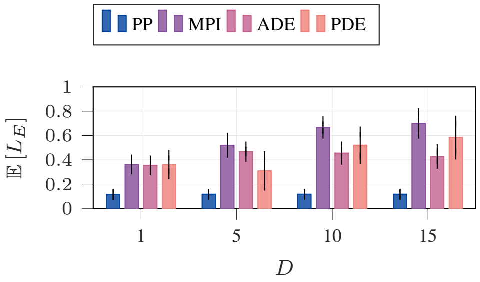
(c) Control reward.

(d) Eve's accuracy.

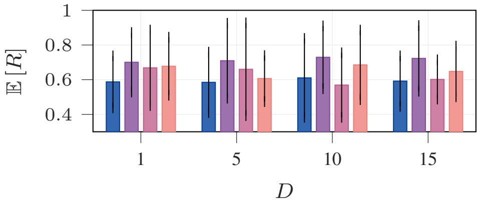
Fig. 16. PDE performance as a function of θ and β in the control scenario, with $D = 5$.

As we can observe from Fig. 15, ADE significantly reduces the accuracy of Eve's estimates, especially in the case of a high transmission rate. A similar effect is achieved using the PDE strategy, whose performance is instead shown in Fig. 16. Since its goal is to avoid L_E exceeding L_{high}, ADE continues to use MPI in many scenarios, especially when $\beta \to 2$ and $\theta \to 2^7$. Instead, PDE decreases the entropy of the scheduling policy in all configurations, modifying the leakage more widely. As shown in Fig. 16a, PDE obtains a lower leakage than ADE, except in scenarios with a low value of both θ and β, in which the initial MPI scheduling is more vulnerable.

In Fig. 17, 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. 17a shows that the optimal GoC scheduling presents an average leakage below 0.6 for $D = 5$, only slightly



(a) Information leakage.



(b) Total reward.

Fig. 17. Expected leakage and reward as a function of the opacity time gap D in the control scenario, with $\beta = 1$ and $\theta = 32$.

higher than the one obtained with ADE. In addition, MPI becomes more vulnerable as D increases, while the average leakage of ADE never exceeds $L_{\text{high}} = 0.6$. The PDE heuristic proves to be more robust than ADE for $D \leq 5$, while leaking more information as the time gap grows. If we consider the total reward, reported in Fig. 17b, the relationship between the two heuristics is inverted: PDE has a degraded performance for $D \leq 5$, while it constitutes an intermediate solution between MPI and ADE for longer time gaps.

D. Pareto Analysis

In the previous analysis, we considered specific hyperparameters for both ADE and PDE, which correspond to a single operation point. In the following, we study the trade-off between secrecy and reward for the two heuristics by computing the Pareto frontier given by all the possible algorithm settings. From a practical perspective, we vary the leakage thresholds $L_{\text{low}} \in [0.1, 0.7]$ while setting $L_{\text{high}} = L_{\text{low}} + 0.2$ in the case of ADE, and the target entropy $H^* \in [0, H(\sigma_{\text{MPI}})]$ in the case of PDE. Importantly, to obtain reliable results, we run a total of $N_{\text{ep}} = 50$ independent episodes per configuration.

Fig. 18 focuses on the remote estimation case: the results show that ADE always outperforms PDE when the leakage is lower than ~ 0.7. This is due to the iterative nature of PDE: 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, which never recovers and is always outperformed by ADE. On the whole, ADE is able to better control the trade-off between secrecy and efficiency in this task, providing an almost linear degradation of the reward as we increase the probability of using PP.

Fig. 19 repeats the 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 while strongly reducing the leakage from 0.5 to values lower than 0.2. On the other hand, ADE seems inefficient in managing the control system and immediately degrades the expected reward. The benefits of PDE are even more relevant as, while being more sophisticated than ADE in its basic mechanism, it does not require Bob to compute the leakage online, thus greatly reducing the computational burden for real-time operation.

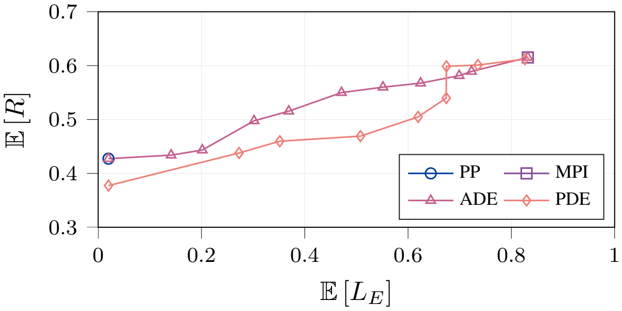


Fig. 18. Pareto frontier of the trade-off between information leakage and reward in the estimation scenario, with $\beta = 1$, $\theta = 32$, and $D = 5$.

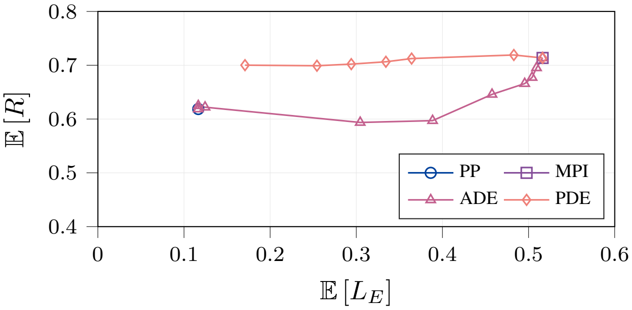


Fig. 19. Pareto frontier of the trade-off between information leakage and reward in the control scenario, with $\beta = 1$, $\theta = 32$, and $D = 5$.

The remote estimation task analyzed in Fig. 18 is characterized by monotonic relations between transition stochasticity, communication cost, and total reward. These trivial performance trends are unlikely in real-world applications, which are expected to be more similar to the remote control task shown in Fig. 19. In scenarios with an irregular relationship between secrecy and efficiency, PDE is much more likely to find solutions that reduce information leakage while preserving the same performance of GoC. In particular, PDE makes suboptimal but more opaque scheduling decisions when this is less critical for the control reward, e.g., when Bob is farther from the target state. Hence, the fact that the control policy is mutually adapted to communication ensure that the system experiences only a negligible performance loss.

VII. Conclusion and Future Work

In this work, we analyzed the security of GoC systems for the remote control of Markov processes, focusing on the system's vulnerability to timing 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., a remote estimation and a remote control problem, and analyzed four different transmission scheduling protocols: the optimal GoC scheduling, a periodic transmission policy, and two heuristic solutions that trade off between the previous approaches.

Our results proved that goal-oriented scheduling has significant performance benefits, but is also highly vulnerable to eavesdropping. In addition, although heuristic mitigation strategies are possible, finding an optimal policy under game-theoretic rationality is a computationally hard problem. We showed that any strategy must be tuned according to the target environment, as the structure of the communication policy may vary significantly depending on factors such as the stochasticity of the system and the transmission 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. Finally, our framework could be applied to push-based scenarios in which Alice independently decides when to send an update, which represents another attractive possibility for future research.

References

[1] C. E. Shannon and W. Weaver, *The mathematical theory of communication*. University of Illinois Press, Sep. 1949.

[2] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang *et al.*, "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE Journal on Selected Areas in Communications*, 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 Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, May 2019.

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

[5] S. Guo, Y. Wang, N. Zhang, Z. Su, T. H. Luan, Z. Tian, and X. Shen, "A survey on semantic communication networks: Architecture, security, and privacy," *IEEE Communications Surveys & Tutorials*, Dec. 2024, Early Access.

[6] T.-Y. Tung and D. Gündüz, "Deep joint source-channel and encryption coding: Secure semantic communications," in *Proc. International Conference on Communications (ICC)*. IEEE, May 2023, pp. 5620–5625.

[7] X. Liu, G. Nan, Q. Cui, Z. Li, P. Liu *et al.*, "SemProtector: A unified framework for semantic protection in deep learning-based semantic communication systems," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 56–62, Nov. 2023.

[8] S. Y. Kung, "A compressive privacy approach to generalized information bottleneck and privacy funnel problems," *Journal of the Franklin Institute*, vol. 355, no. 4, pp. 1846–1872, Mar. 2018.

[9] W. Chen, S. Shao, Q. Yang, Z. Zhang, and P. Zhang, "A nearly information theoretically secure approach for semantic communications over wiretap channel," *arXiv Preprint 2401.13980*, Jan. 2024.

[10] T. Van Goethem, W. Joosen, and N. Nikiforakis, "The clock is still ticking: Timing attacks in the modern Web," in *Proc. 22nd Conference on Computer and Communications Security (CCS)*. ACM SIGSAC, Oct. 2015, pp. 1382–1393.

[11] P. Talli, E. D. Santi, F. Chiariotti, T. Soleymani, F. Mason, A. Zanella, and D. Gündüz, "Pragmatic communication for remote control of finite-state Markov processes," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 7, pp. 2589–2603, Jul. 2025.

[12] F. Mason, F. Chiariotti, P. Talli, and A. Zanella, "Eavesdropping on goal-oriented communication: Timing attacks and countermeasures," in *Proc. 8th INFOCOM Workshop on Age and Semantics of Information (INFOCOM ASoI)*. IEEE, May 2025.

[13] J. Xu, B. Ai, W. Chen, N. Wang, and M. Rodrigues, "Deep joint source-channel coding for image transmission with visual protection," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 6, pp. 1399–1411, Dec. 2023.

[14] Y. Li, Z. Shi, H. Hu, Y. Fu, H. Wang, and H. Lei, "Secure semantic communications: From perspective of physical layer security," *IEEE Communications Letters*, vol. 28, no. 10, pp. 2243–2247, Oct. 2024.

[15] J. Shi, Q. Zhang, W. Zeng, S. Li, and Z. Qin, "Secure transmission in wireless semantic communications with adversarial training," *IEEE Communications Letters*, vol. 29, no. 3, pp. 487–491, Mar. 2025.

[16] Y. Rong, G. Nan, M. Zhang, S. Chen, S. Wang *et al.*, "Semantic entropy can simultaneously benefit transmission efficiency and channel security of wireless semantic communications," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 2067–2082, Jan. 2025.

[17] X. Huang, L. Zeng, Y. Lu, and J. An, "Secure and robust joint source-channel coding with semantic clustering and adversarial purification," *IEEE Transactions on Cognitive Communications and Networking*, Apr. 2025, Early Access.

[18] S. Tang, Y. Chen, Q. Yang, R. Zhang, D. Niyato, and Z. Shi, "Towards secure semantic communications in the presence of intelligent eavesdroppers," *arXiv preprint arXiv:2503.23103*, Mar. 2025.

[19] X. Xu, Y. Chen, B. Wang, Z. Bian, S. Han *et al.*, "CSBA: Covert semantic backdoor attack against intelligent connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 11, pp. 17 923–17 928, Nov. 2024.

[20] M. Shen, J. Wang, H. Du, D. Niyato, X. Tang *et al.*, "Secure semantic communications: Challenges, approaches, and opportunities," *IEEE Network*, vol. 38, no. 4, pp. 197–206, Jul. 2024.

[21] Z. Yang, M. Chen, G. Li, Y. Yang, and Z. Zhang, "Secure semantic communications: Fundamentals and challenges," *IEEE Network*, vol. 38, no. 6, pp. 513–520, Nov. 2024.

[22] R. Meng, S. Gao, D. Fan, H. Gao, Y. Wang *et al.*, "A survey of secure semantic communications," *Journal of Network and Computer Applications*, vol. 239, p. 104181, Jul. 2025.

[23] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam, "Mitigating timing side channel in shared schedulers," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1562–1573, Jun. 2016.

[24] 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. Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, Apr. 2016.

[25] A. Mohaisen, H. Mekky, X. Zhang, H. Xie, and Y. Kim, "Timing attacks on access privacy in information centric networks and countermeasures," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 6, pp. 675–687, Nov. 2015.

[26] L. Mazaré, "Decidability of opacity with non-atomic keys," in *18th World Computer Congress (WCC)*. IFIP, Aug. 2004, pp. 71–84.

[27] 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.

[28] A. Saboori and C. N. Hadjicostis, "Verification of k-step opacity and analysis of its complexity," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 3, pp. 549–559, Jul. 2011.

[29] J. Chen, M. Ibrahim, and R. Kumar, "Quantification of secrecy in partially observed stochastic discrete event systems," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 185–195, Jan. 2017.

[30] S. Liu, X. Yin, and M. Zamani, "On a notion of approximate opacity for discrete-time stochastic control systems," in *Proc. American Control Conference (ACC)*. IEEE, Jul. 2020, pp. 5413–5418.

[31] B. Bérard, K. Chatterjee, and N. Sznajder, "Probabilistic opacity for Markov decision processes," *Information Processing Letters*, vol. 115, no. 1, pp. 52–59, Jan. 2015.

[32] 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.

[33] K. Horák, B. Bošanskỳ, V. Kovařík, and C. Kiekintveld, "Solving zero-sum one-sided partially observable stochastic games," *Artificial Intelligence*, vol. 316, p. 103838, Mar. 2023.

[34] A. Delage, O. Buffet, J. S. Dibangoye, and A. Saffidine, "HSVI can solve zero-sum partially observable stochastic games," *Dynamic Games and Applications*, vol. 14, pp. 751–805, Sep. 2023.

[35] G. B. Dantzig, "A proof of the equivalence of the programming problem and the game problem," in *Activity Analysis of Production and Allocation*. John Wiley & Sons, Jan. 1951, ch. 20, pp. 330–338.