IEEE Vehicular Technology Magazine, Available online 21 July 2023
DOI: 10.1109/MVT.2023.3290773

Control Layer Security: A New Security Paradigm for Cooperative Autonomous Systems

Weisi Guo, Zhuangkun Wei, Oscar Gonzalez, Adolfo Perrusquía, Antonios Tsourdos

***Abstract*—Autonomous systems often cooperate to ensure safe navigation. Embedded within the centralised or distributed coordination mechanisms are a set of observations, unobservable states, and control variables. Security of data transfer between autonomous systems is crucial for safety, and both cryptography and physical layer security methods have been used to secure communication surfaces - each with its drawbacks and dependencies.**

Here, we show for the first time a new wireless Control Layer Security (CLS) mechanism. CLS exploits mutual physical states between cooperative autonomous systems to generate cipher keys. These mutual states are chosen to be observable to legitimate users and not sufficient to eavesdroppers, thereby enhancing the resulting secure capacity. The CLS cipher keys can encrypt data without key exchange or a common key pool, and offers very low information leakage. As such the security of digital data channels is now dependent on physical state estimation rather than wireless channel estimation. This protects the estimation process from wireless jamming and channel entropy dependency. We review for first time what kind of signal processing techniques are used for hidden state estimation and key generation, and the performance of CLS in different case studies.

***Index Terms*—Cybersecurity, wireless communication, control theory, estimation, autonomous systems**

I. Introduction

Autonomous systems (AS) cover a broad range of platforms that have various degrees of autonomy, typically in control (stabilizing movement) and navigation (completing a mission objective). ASs that require control and navigation include but are not limited to autonomous vehicles, aerial drones, robots, and maritime vessels. Typically ASs cooperate together to achieve a common purpose, or have to cooperate because they share a common space (e.g., a road or air corridor). Examples of cooperative ASs include platoon driving, swarm robotics, collision avoidance, and formation flying. In all these cooperative cases, ASs observe each other via direct sensing or data exchange to achieve synchronized behaviors.

A. Review of Cybersecurity

Cybersecurity for wireless communications is essential to secure the knowledge exchange between ASs and with other stakeholders. Examples of wireless data transfer include: sensor data collected by ASs to map an environment, Position-Navigation-Timing signals to ensure safe navigation, and federated gradient knowledge between ASs and a hub. Several wireless security approaches exist and we attempt to summarize them below in different categories. We then differentiate the proposed CLS from them.

1) Cryptography to Post-Quantum Cryptography: Cryptography relies on mathematical and computational complexity to generate and distribute asymmetric/symmetric cipher keys [1]. The challenge lies in the lack of theoretical information theory-based guarantee, as most of the algorithms leverage the complexity of mathematical problems, e.g., the integer factorization problem, the discrete logarithm problem, and the elliptic-curve discrete logarithm problem, of which, however, all could be solved by an eavesdropper (Eve) equipped with a powerful quantum computer [2].

Indeed, post-quantum cryptography is being actively studied to overcome secret keys cracked by a super quantum computer. However, in the resource and computational limited AS network, e.g., the drone network, cryptography-based key computation, and key exchanges will lead to huge computational complexity and communication overhead. Instead, our proposed CLS provides a lightweight pathway to generate symmetric cipher keys, which are more suitable for the AS network.

2) Distributed Ledger Technology: Distributed Ledger Technology (DLT) covers a range of consensus-based security measures suitable for tracking the provenance and usage of data. DLT protocols between ASs generally require either heavy computation (proof-based) or high data exchange overhead (voting-based). As such, they are often unsuitable when the wireless channel is congested, or/and when there is very little delay tolerance (tactile AS controls need subms agreement), and/or when the computational resources are limited on ASs.

3) Physical Layer to Graph Layer Security: Physical layer security (PLS) broadly covers a range of techniques in exploiting physics mechanisms to secure data. This includes the quantum key distribution (QKD), which leverages the quantum mechanisms (e.g., entanglement and indeterminacy) to create linked quantum states of two legitimate parties for shared secret key generation [3]. The limitation of QKD is the high cost of the devices for quantum entanglement and state measuring, and the prerequisite of existing authenticated channels. This then blocks its usage on lightweight ASs, e.g., drone networks.

Another lightweight PLS leverages the physical attributes

This work is supported by the Engineering and Physical Sciences Research Council [grant number: EP/V026763/1].
Weisi Guo, Zhuangkun Wei, Oscar Gonzalez, Adolfo Perrusquía, Antonios Tsourdos are with the School of Aerospace, Transport, and Manufacturing, Cranfield University, MK43 0AL, UK.
Weisi Guo is also with the Alan Turing Institute, London, NW1 2DB, UK.
Corresponding author: weisi.guo@cranfield.ac.uk.

TABLE I
COMPARISON BETWEEN DIFFERENT CRYPTOGRAPHY, PHYSICAL LAYER SECURITY AND PROPOSED CONTROL LAYER SECURITY APPROACHES

	Cryptography	Physical Layer Security	Graph Layer Security	Control Layer Security
Central Security Idea	Use Common Key Pool to Generate Keys	Use Mutual Wireless Features to Generate Distributed Keys	Use Mutual Physical Sensor Data to Generate Distributed Keys	Use Mutual Cooperative AS States to Generate Distributed Keys
Key Generation: Feature Source	Common Key Pool	Wireless Channel	Physical Sensor Reading	Dynamic States of ASs
Key Reciprocity	Guaranteed via Common Key Pool	V. Strong for most EM materials and channels	Strong for connected systems	Strong for cooperative ASs
Key Dynamics & Uniqueness	Limited to Common Key Pool	Depends on channels (Strong for urban areas and mobile, Weak for aerospace)	Depends on demand cycles	Strong for multi-task autonomy
Attack Vectors & Leakage	Key Pool Security determines overall security	Cooperative Eves can estimate legitimate channel, Jamming can decrease channel estimation accuracy	Depends on knowledge of underlying physical network	Unobservable state estimation determines leakage

of the radio channel to secure data. At the most basic level, keyless security can be achieved using dynamic beam steering/forming [4] or drone-controlled reconfigurable intelligent surface (RIS) [5], rotation modulation to distort the constellation, and fingerprinting individual antennas [6]. More recently, advances in generating cipher keys using radio channel properties have enabled more secure channels without relying on a common key pool such as cryptography methods [7]–[9]. However, as PLS derives its security from the very radio channel it is trying to protect, it remains sensitive to jamming, high noise, poor channel entropy or reciprocity, and poor channel estimation quality issues. As such, the limitation of channel based PLS lies in the prerequisite of the reciprocal channel randomness, which will not hold in adversarial scenarios, e.g., jamming, pilot spoofing, and CSI attacks.

Graph Layer Security (GLS) advances PLS to common sensed network states to encrypt digital data [10]. For example, two robots monitoring a sewage network can use commonalities in water flow to generate decentralized cipher keys. This removes the channel estimation dependency of PLS, pushing the burden to physical sensor accuracy. However, ASs do not usually share a common physical network (e.g., water or gas pipelines), and any air-flows between them cannot be a reliable reciprocal source of common physics to generate cipher keys. As such, we must seek other common states to exploit.

B. Gaps in Security Capability for Cooperative ASs

Cooperative ASs operating in close formation often need to make rapid decisions together over a secure channel. There is often a large disparity between the computation capability of an AS platform vs. a powerful premeditated external attacker. This makes current cryptography and DLT approaches not always suitable. These indeed motivated the rapid advances in PLS in recent years, where keyless and key-based PLS have been combined to secure ASs. However, PLS remains sensitive to channel properties and we give some examples where it becomes challenging. For example, (1) airborne platforms typically do not have high entropy channels making PLS keys very static [11], and (2) increasing use of reconfigurable intelligent surfaces (RIS) with non-reciprocal reflection properties can inadvertently undermine or maliciously attack the legitimate PLS secured channel [12].

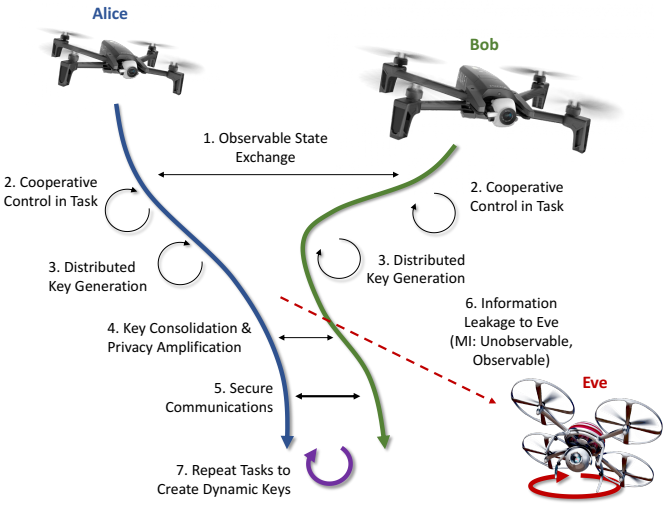

Fig. 1. Control Layer Security: cooperative actions between legitimate ASs (Alice & Bob) allow them to use their states to find common features that can create distributed mutual cipher keys.

C. Contribution and Organisation

The contribution of our work is that this is the first review paper that proposes the idea of control layer security (CLS), which aims to generate mutual control states (common randomness) between cooperative autonomous systems for symmetric cipher key generation. Compared to QKD and PLS, CLS is premised on the control layer cooperation, and thereby does not require either extra devices and a secured channel (QKD needed), or the reciprocal channel randomness (PLS needed), which are hard to be satisfied by lightweight ASs in an adversarial environment.

The paper is organized in the following sections:

1) Review and comparison of related technologies ranging from cryptography, physical layer security (PLS), and graph layer security (GLS);
2) Introduce the control theory, signal processing, and communication theory mechanisms of CLS - including observable and unobservable autonomous system states, feature extraction, and key generation;
3) Demonstration of CLS across different autonomous system mission tasks and environments, with different

eavesdropper attack scenarios;
4) Discussion of CLS and future research directions;

II. Control Layer Security of Autonomous Systems

Control layer security requires the legitimate nodes (ASs) to be cooperative in the control layer, so that they can have mutual states for symmetric cipher key generation. The cooperative control exists for a wide range of tasks, e.g., rescuing searching, platoon driving, formation flight, swarm tasking...etc, which are the potential scenarios in which CLS can generate cipher keys to secure their communication streaming. An overall algorithm flow of CLS is shown in Figure 1. In this section, we show that both simple to complex cooperative control (from linear controllers to reinforcement learning) can enable ASs to have mutual states. Some of these mutual states are unobservable to an outside eavesdropper and only observable to the ASs themselves. We begin with an introduction to dynamic AS models (with controllers) and state feature extraction.

A. Observable and Unobservable States

In general, dynamic systems can be expressed as an evolution equation of states $\mathbf{X}$:

$$\mathbf{X}_{k+1} = \mathbf{A}\mathbf{X}_k + \mathbf{B}\mathbf{U}_k, \mathbf{Y}_k = \mathbf{C}\mathbf{X}_k; \quad (1)$$

where $\mathbf{A}$ is the evolution matrix that models the dynamics of the vehicle, and $\mathbf{U}$ is an autonomous control signal with transformation matrix $\mathbf{B}$. Within the state vector $\mathbf{X}$, there can be a range of observable and unobservable states (examples given in Table II and visualization in Figure 2). Here, we can see that depending on the sensor used and the range of the state observation channel, there is a range of: (1) directly observable, (2) partially observable (requires noisy inference/calculation), and (3) unobservable states. For close formation flight between cooperative ASs, the state observation channel between legitimate ASs is short and LoS-dominated. However, the channels between some Eves and Alice/Bob can be much longer and possibly NLoS based. Therefore, we can potentially select visual sensors between Alice and Bob using Position, Roll, Pitch, and Yaw as observable states; and knowing that Eve using Radar or Radio techniques cannot (easily) observe these states.

B. Cooperative Control

For cooperative control, other ASs must have visibility of certain states. In this case, we also define only observable states as $\mathbf{Y}$, which is a transformation of the $\mathbf{X}$ via the observation matrix $\mathbf{C}$.

For CLS, legitimate ASs should be cooperative in the control layer. They either have a centralized controller with combined objective function (e.g., cooperative control), or have distributed controllers in their ends involving others' observable states. The cooperative design aims to generate highly correlated unobservable states at Alice and Bob (two legitimate ASs). This can be achieved by adding to their objective functions the cooperative components which involve other's observable states (e.g., the x, y, z positions). In this way, they will have correlated but unobservable states (e.g., their yaw angles are unobservable due to the geometric shape of a quadcopter). The symmetric cipher keys can then be generated at their ends using the correlated states.

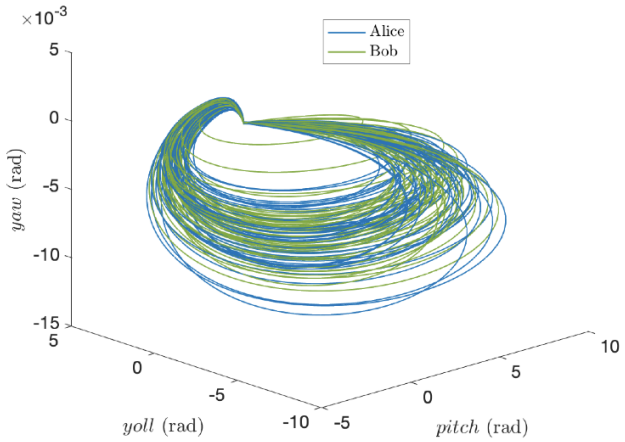

Fig. 2. Example of Alice and Bob flight trajectory's resulting state space evolution that acts as features for CLS cipher key generation.

It is noteworthy that the objective functions can be the ones from the linear quadratic regulator (LQR), the model predictive control (MPC), and also the reward functions in deep reinforcement learning, all of which can be combined with the cooperative control components for generating correlated states for symmetric key generations.

1) Formation Control: In classic formation control, for the cooperative control between Alice a and Bob b, we are interested in designing reward functions J such that we maximize the joint reward between (see Figure 3a):

1) task completion measured by the states of Alice $\mathbf{X}^{(a)}$
2) mutual task completion measured by the joint states of Alice $\mathbf{X}^{(a)}$ and Bob $\mathbf{Y}^{(b)}$

Here, the reward function in ASs is time-varying from one micro-task to the next (e.g., following lanes, overtaking vehicles, collision avoidance...etc.). At the east task, we can then optimize J per time step using a variety of autonomous controllers, varying from linear control (e.g., LQR, proportional–integral–derivative controllers), to reinforcement learning control.

2) Model Predictive Control: A popular approach to tackle the task of cooperative and/or formation control is Model Predictive Control, in which at each time step, it optimizes the future control actions (and as a result, their trajectories and observable states $\mathbf{Y}$) of each agent according to the cost function. Several variants of this type of control have been studied, focusing mostly on centralized, decentralized, and distributed solutions for UAV Swarms, with the latter being one of the most dominant given its enhanced performance when compared to its decentralized counterpart, as well as its modular/reliable framework when compared to the centralized solution. This allows a stronger long-term correlation of the

TABLE II
STATE CHANNEL MODEL: OBSERVABLE AND UNOBSERVABLE STATES FOR AUTONOMOUS SYSTEMS (ASS)

	Range & Channel	Observable States	Partially Observable States (Calculated)	Unobservable States
Ultrasonic Sensor (kHz)	V. Short (m), LoS	Position	Velocity	Acceleration, Roll, Pitch, Yaw
Visual / IR Sensor (THz)	Short & Medium (km), LoS	Position, Roll, Pitch, Yaw	Velocity	Acceleration
Radar / Lidar (6-300 GHz)	Medium to V. Long (<1000 km), LoS	Position, Velocity	Acceleration	Roll, Pitch, Yaw
Radio Transponder (27-2400 MHz)	Long (100s km), NLoS	Position	Velocity	Acceleration, Roll, Pitch, Yaw

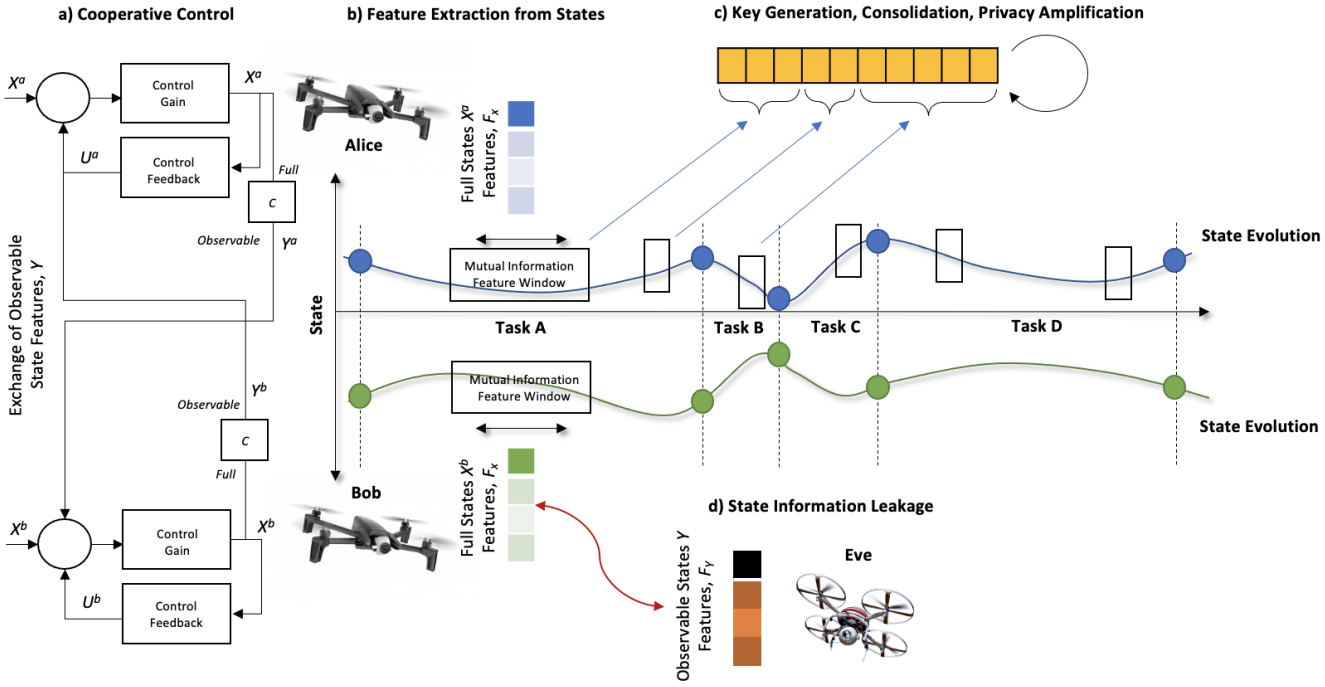

Fig. 3. Control Layer Security: (a) cooperative control between legitimate ASs (Alice & Bob) uses observable features $\mathbf{Y}$, which is a subset of full features $\mathbf{X}$; and (b) cooperative control enables correlated $\mathbf{X}$ that in turn can use a sliding window to identify strongest features F_X that (c) create reciprocal cipher keys between Alice & Bob. (d) This in turn has a low leakage rate to Eve as Eve can only obtain features from observable states F_Y in different types of attack scenarios.

full-state features $\mathbf{X}$ between Alice and Bob, thus ensuring stronger cipher key generation.

An important assumption for the distributed framework is that the agents exchange partial (or in some cases complete) information about their states and trajectories which in turn allows the other agents to become aware of the current trajectories that are being planned. This results in consistency between the optimization problems solved by the various agents, which in turn improves the convergence and stability of the overall solution, and therefore of their correlated actions. At this point, it should be mentioned that MPC is often used as a "high-level" planner where only virtual control variables are used, as discussed in [13]. Thus, by exchanging only these high-level control actions, prevents the estimation of the full state as discussed later in section III-B3, which in turn allows the use of the proposed framework.

3) Game Theory-Based Control: Another effective way to model the cooperation between multiple ASs is by means of game theory techniques. In game theory, the interaction between agents' behaviors is determined by a set of local decisions based on partial information of each other. The cooperative architecture depends on the nature of the problem and can be modeled either as multiplayer games or multi-agent interactions where several players (control inputs) or agents (e.g., Alice and Bob) interact with each other to optimize different objective functions in either a centralized or distributed plant. A complete explanation of these techniques can be found in [14].

One common feature of the game theoretical control formulations is the assumption that each agent has access to the full state of the system, that is, $\mathbf{C}$ is an identity matrix of appropriate dimension. However, novel reinforcement learning architectures have been developed for a single agent to obtain optimal control policies using only partial states measurements Y [15]. The only condition to verify is that the observability

matrix

$$\mathcal{O} = [\mathbf{C}^\top, (\mathbf{CA})^\top, ..., (\mathbf{CA}^{K-1})^\top]^\top \tag{2}$$

has full rank. Otherwise, the unobservable states must be stable or, in other words, the AS is detectable.

C. Feature Extraction

All legitimate ASs must have a feature extraction algorithm to translate states $\mathbf{X}$ into features F_X that can be used for key generation. Like PLS and GLS, the CLS-based control layer features from states that meet several criteria simultaneously:

1) Reciprocity: Here, the states of Alice and Bob must be correlated via the aforementioned cooperative control. Not all states that exist need to be reciprocal, as some cannot be observed or are not used in cooperative control. Feature extraction from states must therefore leverage those states that are reciprocal. CLS does not need their referenced trajectories to be correlated for the symmetric key generation, as their state feature correlation is generated by cooperative control designs.
2) Dynamic and Random: Here, the states must be time-varying and control a degree of randomness in order to avoid Eve successfully performing a brute-force attack. In reality, what this means for the ASs, is that we can naturally expect noise in 3 forms: (1) dynamic change of environment, (2) observation noise in $\mathbf{C}$, and (3) reward function variations from task to task. Note that the randomness will not be compromised even with repeated tasks, where their real-time states are still with randomness given the time-varying environment (e.g., noisy IMU, gyro, and GNSS measurements, dynamic dragging effect, etc).
3) Uniqueness: Here, we assume that given a long run of sequential tasks between a group of ASs, this sequence and the states within these ASs (given the above random conditions) make it unique.
4) Low State Leakage: Whilst the above 3 requirements ensure successful and high-quality key generation, a strong eavesdropper (Eve) can still observe these states/features. In order to reduce the key leakage rate, we must ensure that we use observable states $\mathbf{Y}$ such that the leakage between $\mathbf{Y}$ and $\mathbf{X}$ is minimized. In the cooperative control case, we do so by minimizing this leakage via both the selection of states via $\mathbf{C}$, as well as the time-varying reward function J design.

For the actual feature extraction, one of the ways one can do it is to extract the state vector $\mathbf{X}$ from Alice and Bob in a distributed manner. One of the challenges unique to ASs, is that the trajectory dynamics can be very smooth sometimes, leading to long runs of state values with very little change. One way to solve this problem is by embedding sequential states as a matrix and calculating the mutual information (MI) such that the intrinsic value of features around state $\mathbf{X}_{k-N/2}$ is evaluated (see Figure 3b):

$$\text{MI}\left(\begin{bmatrix}\mathbf{X}_k^{(a)}\\ \vdots \\ \mathbf{X}_{k-N}^{(a)}\end{bmatrix}; \begin{bmatrix}\mathbf{X}_k^{(b)}\\ \vdots \\ \mathbf{X}_{k-N}^{(b)}\end{bmatrix}\right), \tag{3}$$

and the state sequences with the highest MI are used for features.

Indeed the non-legitimate ASs could have the feature extraction algorithm, but their states (as the inputs for the feature extraction algorithm) are not correlated with the legitimate pair, as the legitimate pair does not involve the states of non-legitimate ASs in their controller. Also, the analysis on the possibility to steal/estimate the unobservable of legitimate ASs are provided in Section III B.

The complexity of the feature extraction algorithm depends on how correlated the states of the legitimate pair are. For the result in Fig. 4, the correlated states of two UAVs are their yaw angles, which are extracted directly as the common features. So, in this way, a UAV only equipped with basic IMU and gyro sensors can easily extract its yaw angle as the common feature, which has been ensured as correlated by the cooperative controller design.

D. Key Generation

After feature extraction, we can create the symmetric secret keys as per traditional PLS techniques - see Figure 1 and 3c. This is namely the steps of: key consolidation and privacy amplification, secure communications, and then following tasks with new or repeated references. This is all whilst dealing with the unique challenges of AS dynamics, namely: (a) cooperative control has transition phases that constantly revolve around stable and unstable regimes, where the degree of correlation of states and features is closely related to mutual stability, and (b) the observation sampling must be sufficiently high to capture the state evolution.

For CLS key generation process, there is no need for key exchange. The state exchange is the basic step for distributed cooperative controls (e.g., model predictive control MPC), to ensure the awareness of others' observable states by each AS. This state-sharing process indeed leads to extra communication overhead in an AS network, but there is no difference with existing cooperative control regulations/protocols, e.g., Automatic Dependent Surveillance–Broadcast (ADB-S).

III. Control Layer Security (CLS): Performance and Demonstration

A. Information Leakage between Unobservable and Observable States

As mentioned previously, the information shared between legitimate ASs Alice and Bob are their mutually observable states $\mathbf{Y}$, which we assume is also observable by Eve $\mathbf{Y}_k^{(e)}$ (even if Eve is further away and/or have greater observation noise). What we show is that the mutual information (MI) between legitimate users presented in Equation 3 is much stronger than between Eve and legitimate users. This is due to 2 reasons embedded in the design of CLS:

1) Information Loss in Observation Matrix $\mathbf{C}$: Eve cannot recover all states $\mathbf{X}$ from observable states $\mathbf{Y}$ as the observation matrix $\mathbf{C}$ is not reversible. This is true even with the AS dynamic model and the control model is known, as all the real-time values of $\mathbf{X}$ are not known to Eve;

2) Information Loss in Cooperative Control Reward J: Eve cannot recover all states $\mathbf{X}$ from observable states $\mathbf{Y}$ even with the AS dynamic model and the control model as the controller is designed such that $\mathbf{Y}$ only partially contributes to future values of $\mathbf{X}$.

As such, we can see from Figure 3d that a low leakage rate to Eve is achieved as Eve can only obtain features from observable states F_Y in different types of scenarios discussed below.

B. Attack Vectors and AS Scenarios

After the elaboration of the CLS-based secret key generation, we study how secure the proposed key is, in the face of different types of Eves. Here, the Eves aim to reconstruct the control-layer common features of Alice and Bob, i.e., F_X, and then regenerate their cipher keys based on these features. To evaluate the security performance of the proposed CLS-based secret key, we consider three types of Eves, with the increase of the knowledge of Alice's and Bob's observable states and systems.

1) Brute-Force Eve (Type-1): The brute-force Eve is assumed to be the simplest Eve without any knowledge of Alice's and Bob's systems, i.e., Eq. (1), nor their observable states $\mathbf{Y}$ (e.g., 3D positions and velocities in Table. II). In this case, the control-layer extracted common features at Alice and Bob cannot be estimated by Eve, and so do the generated cipher keys relied upon.

2) Model-Free Eve (Type-2): Model-Free Eve refers to Eves that can obtain the observable states of Alice and Bob $\mathbf{Y}$ (e.g., the 3D positions and the speeds shown in Table. II). In this case, Eve will use the observed states as the legitimate features to reconstruct the cipher keys. As stated in the feature extraction part (Section II. C. 4)), minimizing leakage between observable and unobservable states serves as the vital solution to prevent such observable state-based Eve. This can be pursued (naturally & artificially) by the following 3 aspects:

1) (Passively) The correlation coefficient between Eve's observed states $\mathbf{Y}^{(e)}$ and Alice's and Bob's state $\mathbf{X}$ is naturally weakened by the observing noise, even worsen with the increase of observing distance from legitimate ASs to Eve, as it induces larger estimation error from noisy observations.
2) (Passively) The information loss from Eve's observed state $\mathbf{Y}^{(e)}$ to legitimate state $\mathbf{X}$ further reduces and its determined secret key leakage rate. For example, there exists a multiple-to-one mapping from the unobservable pitch, roll, and yaw angles to Eve's observed trajectories (e.g., going forward can be pursued either by direct pitch angle controlling or by clockwise yawing and rolling). This renders the difficulty for Eve to obtain the legitimate secret keys leveraging Alice's and Bob's unobservable yaw states.
3) (Artificially) As is expressed the correlation of Eve's observed and legitimate states, the idea to minimize is to artificially and jointly optimize (i) the observing matrix, and (ii) the cooperative reward objectives J to minimize the selected entries of the covariance matrix.

3) Kalman Filter Eve (Type-3): We next consider a strong Eve with (i) the knowledge of AS's dynamic & control model, i.e., Eq. (1), and (ii) the observable states of Alice and Bob. Notably, these assumed prerequisites of Eve are extremely strong (even if guessing the modeling and intention is a separate research flow), but we will show that even so, Eve still cannot estimate the CLS-based secret key generated at Alice and Bob.

From Eve's perspective, the derivation of the secret key can be converted to estimate Alice's and Bob's dynamic states via the observed states. This can be generally pursued by (i) estimating the initial state, and (ii) taking the estimated initial states and observations into the sequential state estimation algorithms (e.g., Kalman filter or Bayesian filter) for further state acquisition.

The difficulty of the Kalman filter-based Eve lies in the multiple-to-one mapping from the initial states to the observations, rendering the under-determined problem given by the observability matrix. An intuitive example is that there are multiple combinations of unobservable yaw, pitch, and roll angles that can map to the same AS's trajectory (which is observed by Eve), making her difficult to estimate them via the observed trajectory. This, on the other hand, provides an insight to defend Type-3 Eve, i.e., the design of the control signal should make the conditional number of $\mathcal{O}$ large.

C. Security KPIs & Overhead Metrics

The evaluation process includes the security KPIs, the overhead analysis, and the computational analysis. The security KPIs are as same as PLS and QKD, given that they all deal with cipher key generation. These include the secret key rate (SKR) as the difference between the mutual information of the common features to the leakage entropy, the key disagreement rate (KDR) before reconciliation, and the p-value for the cipher key randomness check.

The main overhead metrics are from the cost of the controller's input, i.e., $\mathbf{U}_k^T\mathbf{U}_k$ in Eq. (1) to illustrate how large this value is compared to the controller without CLS. The next metric will be the store and computational complexity for CLS common feature extraction. Then, other overhead costs are the same as the PLS and the QKD, for key quantization, key reconciliation and privacy amplification.

The computational complexity contains the complexities of (i) CLS to provide common states, (ii) key quantization from common features to binary key, (iii) key reconciliation, and (iv) privacy amplification process. Here, all steps of (ii)-(iv) are the same as those of PLS and QKD, and therefore with the same computational complexity. The computational complexity analysis of (i) depends on which cooperative controller is used. For example, linear controllers (e.g., LQR) can provide explicit expressions of the input control variables, i.e., $\mathbf{U}_k$, whose complexity is mainly spent on the computations of the Riccati function. The MPC that requires solving the constrained quadratic objective function in each controlling time step should be considered as what algorithms are used (e.g., interior point method). The deep reinforcement learning (DRL) with already trained actor neural network should be

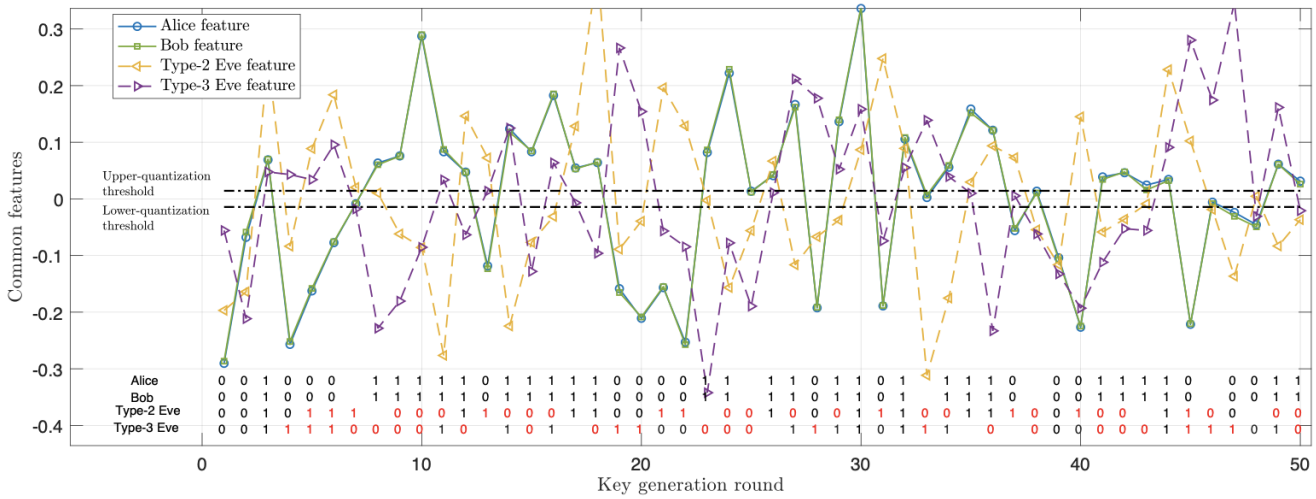

Fig. 4. Control Layer Security in action: key generation over time with optimal feature quantization to ensure strong secrecy performance against Type-2 and Type-3 Eves.

considered by the number of neurons and layers and the exact complexity of the activation function. Ongoing training onboard DRL should be considered as which DRL is used (e.g., DQN, DDPG, PPO) and the on/off policy for training.

D. Example Results of CLS in Action

Here in Figure 4 we show how keys are generated from common features between Alice and Bob during a real formation flight (visualized in Figure 2). For the test set in Fig. 4, Alice and Bob are set as two geometrically symmetrical quadcopters. The state vector of each quacopter is the stacked 12 states, i.e., the xyz positions, the xyz velocities, roll, pitch, yaw, and the rates of roll, pitch and yaw. The dynamic and control model is the widely used linearized quadcopter model. We assume that the 3D xyz-positions can be observed with observing errors. And the corresponding xyz-velocities can be computed via the differences. For the control signal design of each UAV (e.g., Alice), we design the cooperative term via Alice's whole states and the observed 3D xyz-positions of Bob. Then, this cooperative term is added to the LQR objective function, which by solving, will derive the control signals that lead to correlated yaw angles (unobservable by others given the geometrical symmetry) between Alice and Bob. Then, the key generation step is pursued by the optimized upper and lower quantization thresholds.

It is illustrated that the features and cipher keys generated by Alice and Bob share both commonality and randomness, rendering the difficulty for a brute-force Eve to guess/estimate. Then, in Figure 4, we test the proposed CLS in the face of both Type-2 (model-free) and Type-3 (Kalman-filter) Eves. It is shown that neither of them can successfully guess/infer the cipher keys, despite Type-3 having the predictive flight physics model of Alice and Bob.

The reasons behind the security performance of our proposed CLS are categorized into three aspects. First, the common features between Alice and Bob are induced by cooperative control, which creates the involvement of each other's states and leads to highly correlated states for further cipher key generation. Second, for Type-2 Eve that tries to steal the legitimate common features by Alice's and Bob's observable states, the small information leakage to the selected and unobservable states of Alice and Bob gives rise to the low correlation between the features of Type-2 Eve and legitimate ASs. Third, for the Kalman-filter-based Eve with knowledge of dynamic & control model and Alice's and Bob's observable states, the under-determined challenge to estimate the initial state from the observable states prevent it from obtaining the legitimate features and further the cipher keys relied upon. As such, the results provide a first glance on the potential of the concept of CLS to secure the communications of ASs, which serves as a promising candidate in scenarios where PLS and GLS cannot hold their prerequisites.

The time consumption to process CLS is within the control interval, i.e., $0.02s$ in our simulation. In this way, the raw secret key rate is proportional to $0.02s$, which, although slower than the data communication rate, can be increased by privacy amplification techniques (which also serve as the last step for other symmetrical key generation, e.g., PLS, QKD).

IV. Future Works

A. Compatibility with Widely Used Controllers

Future work will focus on other forms of cooperative control discussed in this paper such as Deep Reinforcement Learning and Game Theoretic Control, but also consider other state features which can offer lower leakage rates to Eve. We will continue also Eve's research to develop new attack vectors, and build physical demonstrators to showcase to the IEEE community.

B. Scalability Issue

How scalability affects CLS may require further studies. Currently, the CLS-based key is an end-to-end cipher key,

generated by the cooperative control of two legitimate nodes (Alice and Bob). In this view, whether the involvement of other legitimate nodes into the cooperative control framework affects the Alice-Bob cipher key will be a worthwhile study. Another interesting topic would be how to generate CLS-based group cipher keys to encrypt the networked communication streaming. This may require more sophisticated control designs to generate unobservable states that possess high correlations by a group of legitimate nodes.

C. Authentication Challenges

It is noteworthy that the concept of CLS cannot replace the authentication process. For example, if a spoofing Carlos aims to pretend as Alice, then Bob and Carlos will use each other's observable states for CLS process, and this leads to the result that Bob will have the symmetric secret keys with Carlos other than Alice. Nevertheless, CLS can be used to help the identification process. For instance, instead of observing the observable states of each other, legitimate Alice and Bob can broadcast their observable states encrypted by their previously generated CLS-based cipher keys. In this way, they will receive the correct observable states of each other, other than the states of the spoofing Carlos who does not access to the CLS keys to encrypt his spoofing states.

D. Information-Theoretic Security

One critical branch of the future work lies in the design and proof of the information-theoretic security of CLS (i.e., unconditional security). For currently, the CLS is implemented on the cooperative control of the geometric symmetry quadcopters, whose yaw angles are hard to be estimated by GNSS or imaging-based Eves, and thereby serve as the unobservable states for cipher key generation. Indeed, if Eve is very close to one legitimate quadcopter, it may be possible to estimate the changes in the yaw angle by image processing techniques, which may break the information-theoretic security. This should be further studied especially via real experiments, by taking into account the image resolution, the sampling time-interval, and the physical safe distance (for now we are using the air-gear 450 quadcopter, which does not allow any object to be close at 1m or there will be a destroy of the propellers).

V. Discussions and Conclusions

The rapid advances in networked ASs working in cooperative formations with data sharing mean new cybersecurity methods have to be developed for them. When we consider existing cryptographic methods, we are naturally concerned by superior computational attackers and there have been many cases of this. This post-quantum cryptography issue indeed motivated the rapid advances in PLS in recent years. However, PLS remains sensitive to attackers (e.g., jamming, pilot spoofing) that destroy its prerequisite wireless channel properties (e.g., reciprocity challenged by new intelligent meta-surfaces). In this review, we propose a new cybersecurity mechanism called control layer security (CLS). The idea of CLS is to exploit the correlated and unobservable states between cooperative ASs to generate cipher keys.

We demonstrated this idea with a pair of UAVs using cooperative formation flight. We showed that even if Eve has full knowledge of observable states and systems cannot estimate the unobservable states and the secret key relied upon, due to the multiple-to-one mapping from unobservable states (pitch, roll, and yaw angles) to the observable states (3D trajectory). This demonstrates a promising candidate to secure the communications of ASs, especially in the adversarial radio environment with attackers that destroys the prerequisite for current PLS schemes.

References

[1] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC press, 2020.

[2] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.

[3] Y. Zhang, Z. Li, Z. Chen, C. Weedbrook, Y. Zhao, X. Wang, Y. Huang, C. Xu, X. Zhang, Z. Wang *et al.*, "Continuous-variable qkd over 50 km commercial fiber," *Quantum Science and Technology*, vol. 4, no. 3, p. 035006, 2019.

[4] X. Pang, N. Zhao, J. Tang, C. Wu, D. Niyato, and K.-K. Wong, "Irs-assisted secure uav transmission via joint trajectory and beamforming design," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 1140–1152, 2022.

[5] Y. Cao, S. Xu, J. Liu, and N. Kato, "Toward smart and secure v2x communication in 5g and beyond: A uav-enabled aerial intelligent reflecting surface solution," *IEEE Vehicular Technology Magazine*, vol. 17, no. 1, pp. 66–73, 2022.

[6] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A review of radio frequency fingerprinting techniques," *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 3, pp. 222–233, 2020.

[7] B. M. ElHalawany, A. A. A. El-Banna, and K. Wu, "Physical-layer security and privacy for vehicle-to-everything," *IEEE Communications Magazine*, vol. 57, no. 10, pp. 84–90, 2019.

[8] Q. Wu, W. Mei, and R. Zhang, "Safeguarding wireless network with uavs: A physical layer security perspective," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 12–18, 2019.

[9] Z. Wei, B. Li, and W. Guo, "Adversarial reconfigurable intelligent surface against physical layer key generation," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2368–2381, 2023.

[10] Z. Wei, L. Wang, S. C. Sun, B. Li, and W. Guo, "Graph layer security: Encrypting information via common networked physics," *Sensors*, vol. 10, no. 3951, 2022.

[11] P. Staat, H. Elders-Boll, M. Heinrichs, R. Kronberger, C. Zenger, and C. Paar, "Intelligent reflecting surface-assisted wireless key generation for low-entropy environments," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2021, pp. 745–751.

[12] Z. Wei, W. Guo, and B. Li, "A multi-eavesdropper scheme against ris secured los-dominated channel," *IEEE Communications Letters*, vol. 26, no. 6, pp. 1221–1225, 2022.

[13] O. J. Gonzalez V and A. Tsourdos, "A laguerre-based distributed nonlinear model predictive control scheme for dynamic obstacle avoidance on multi-rotor uavs," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 1632–1637.

[14] K. G. Vamvoudakis, H. Modares, B. Kiumarsi, and F. L. Lewis, "Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online," *IEEE Control Systems Magazine*, vol. 37, no. 1, pp. 33–52, 2017.

[15] A. Perrusquía, "Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning," *Information Sciences*, vol. 595, pp. 364–377, 2022.

2023-07-21

Control layer security: a new security paradigm for cooperative autonomous systems

Guo, Weisi

IEEE

Guo W, Wei Z, Gonzalez O, et al., (2023) Control layer security: a new security paradigm for cooperative autonomous systems, IEEE Vehicular Technology Magazine, Available online 21 July 2023

https://doi.org/10.1109/MVT.2023.3290773