A graph-theoretic understanding of network-wide implications of local cyber protections for mission assurance

Sean Lane[a], Michael R. Clark[b], Mengran Xue[c], Sandip Roy[c], and Sean Warnick[d]

[a]*seanlane@byu.edu*, Brigham Young University, Provo, UT, USA
[b]*mclark@riversideresearch.org*, Riverside Research, Beavercreek, OH, USA
[c]{*sroy,mxue*}*@eecs.wsu.edu*, Washington State University, Pullman, WA, USA
[d]*sean@cs.byu.edu*, Brigham Young University, Provo, UT, USA

ABSTRACT

Modern Department of Defense mission systems are very complex and therefore arduous to defend, especially in the cyber domain. A major cause for this concern arises from the fact that implementation of security protections occurs at a local scale, while the important operational security issues stem from a global perspective of the system, e.g. mission assurance. Being able to understand network-wide implications of local cyber protections has the potential to significantly impact the strategies we use to protect modern mission systems. In this work, we present a graph-theoretic perspective on this problem, which is based on a framework for modeling and assessing the integrated cyber-physical dynamics of complex systems. Under the framework, these dynamics (and their relationships) are modeled as a graph and then analyzed using processing techniques from graph-theory. We demonstrate the utility of this framework by conducting insider-attack threat analysis and show how the application of security protections at a local scale impacts network-wide security properties from an insider perspective.

As a test case, we study the problem of search-and-rescue (SAR) using unmanned aerial vehicle teams. Unmanned vehicle teams engaged in SAR are prototypical cyber-physical systems, in which local intrusions may cause global disruptions. Here, we describe how the insider modeling framework for cyber-physical dynamics applies to this problem, and present results of a network-wide assessment of security properties of the system. We use this assessment to design a security protection for the system in which we use cryptographically secure computation techniques to limit the amount of information sharing required between system components without degrading correct operation of the system. We show how the application of these techniques on a local-scale impacts the security properties of the system on a global-scale.

Keywords: cyber security, graph-theory, insider threat, multiparty computation, simulation, state inference

1. INTRODUCTION

Complexity of modern systems leads to an inability to fully understand how compromises can be leveraged by an adversary to impact mission assurance. This leads to the inability to optimize protection of systems, which results in wasted effort and increased costs in cyber defense.

We show that a graph-theoretic framework for modeling and assessing the integrated cyber-physical dynamics of complex systems can be used to inform a system protection strategy, even in the face of "insider" attackers who have prior understanding and knowledge of the system structure. We do this through the use of a SAR example using unmanned aerial vehicle teams. We apply the framework to the example problem, use the results of the assessment to apply a specific protection, cryptographically secure computation, to limit the amount of information sharing required by the team. We then show how the application of the protections, which happen on a local-scale, impacts the security properties of the system on a global-scale.

Protection strategies followed today are reactionary. The analytical, quantitative approach breaks the cat-and-mouse paradigm we consistently see in cyber defense. This should reduce costs associated with protections because it allows for the prioritization for where limited defensive resources are allocated and results in more efficient investment in the cyber security of complex systems.

2. THREAT MODEL

Many cyber security implementations rely on existing security infrastructure, protocols, and practices to maintain system integrity, but due to expense and difficulty, security is not designed into the structure of the system itself. We assume there is an unmanned aerial vehicle (UAV) swarm consisting of a number of UAVs that are cooperating in a search and rescue mission over an identified track or pattern.

The attack being modeled in this scenario will be observation, inference, or spying attacks where the adversary is attempting to learn the position and velocity information which the swarm members share to coordinate their flight paths that could lead to harmful action against our UAV swarm or other friendly forces. Examples of possible consequences to the release of this information are the members of the swarm being shot down, the general location of the target of the search and rescue operation being inferred from the search pattern of the swarm, or the location of the home base of each swarm member being inferred from the tracking data.

Our adversary is assumed to have breached any existing security infrastructure regarding one or more of UAVs and can actively listen to the information being sent and received by each compromised UAV. She understands the objective of the swarm in as much as the swarm is participating in a search and rescue operation, and she intends to utilize this information in an adversarial manner to the detriment of the swarm or friendly forces.

3. MODELING FRAMEWORK

As part of our case formulation, we will define a modeling framework that describes how the individual UAVs will coordinate and interact with the other members of the swarm. Literature review shows a number of peer-reviewed methods of modeling UAVs cooperating in a shared task. These include using Dubins path generation techniques[1] on fixed-wing UAVs,[2] consensus dynamics within adjacency graphs,[3] and first principle's based double-integrator-network (DIN) models.[4] Our model is based on the previous research by Xue, et al. (2014),[5] which utilizes the DIN model and describes the inherent security of the swarm to spying attacks through classical definitions of system observability. This framework enables the abstract representation of the agents of the swarm, illustrating the cyber and physical capabilities of the members to communicate and collaborate on an assigned tracking task. Here we will review the DIN and adversary models used in this case example.

3.1 The DIN

We will consider a team of n UAVs labeled as $i = 1 \dots n$, with each vehicle incorporating dynamics individually and as a part of the overall swarm. These dynamics can be described through a state space representation that incorporates the multidimensional status of the vehicle's physical position and velocity. The classical state space representation prescribes matrices A, B, C, and D that describe the effect of the current swarm state on the future state, the input on the future state, the current state on the output, and the input on the output, respectively. For our purposes, we can assume that $D = \mathbf{0}$ for the remainder of this paper.

With this model, we can simulate any individual agent i with the equations

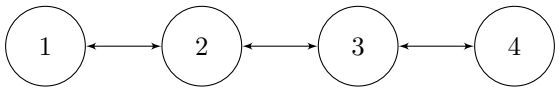
$$\dot{\boldsymbol{x}}_i = A_i \boldsymbol{x}_i + B_i \boldsymbol{u}_i \qquad \boldsymbol{y}_i = C_i \boldsymbol{x}_i \tag{1}$$

where $\boldsymbol{u}_i, \boldsymbol{x}_i, \boldsymbol{y}_i$ represent the individual vehicle input, state, and output vectors. We can then combine the matrices pertaining to each vehicle and merge them into a system wide state space representation with the respective vehicle matrices forming the block diagonal system matrices

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix} \triangleq \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \vdots \\ \dot{\boldsymbol{x}}_n \\ \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_n \end{bmatrix} = \left[\begin{array}{ccc|ccc} A_1 & & \mathbf{0} & B_1 & & \mathbf{0} \\ & \ddots & & & \ddots & \\ \mathbf{0} & & A_n & \mathbf{0} & & B_n \\ \hline C_1 & & \mathbf{0} & & & \\ & \ddots & & & \mathbf{0} & \\ \mathbf{0} & & C_n & & & \end{array}\right] \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_n \end{bmatrix}. \tag{2}$$

Another assumption we make is that the UAVs must communicate and coordinate in order to achieve a cooperative mission objective, which we can model through the individual C_i matrices. For example, suppose we have a network of 4 drones as depicted in Figure 1 where the swarm is organized in a linear formation such that drone 1 interacts with drone 2, drone 2 with drones 1 and 3, drone 3 with drones 2 and 4, and drone 4 with drone 3.

(a) Illustration of example drone formation for search and rescue.



(b) Abstract information graph corresponds to the formation depicted in Figure 1a. Note that the analysis techniques used in this work allow for arbitrary team formations; this simple structure is used here for pedagogical clarity.

Figure 1: Example UAV Team Configuration

In this scenario, the C matrix of the total system is set so that drones not only measure their own individual states, but those of the connected neighbors. This implies that the individual C_i matrices each end up being a form of the identity-zero matrix $\begin{bmatrix} I & 0 \end{bmatrix}$, which for this case would be

$$C_1 = \begin{bmatrix} \boldsymbol{I} & 0 & 0 & 0 \\ 0 & \boldsymbol{I} & 0 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} \boldsymbol{I} & 0 & 0 & 0 \\ 0 & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & \boldsymbol{I} & 0 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 0 & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & 0 & \boldsymbol{I} \end{bmatrix}, \quad C_4 = \begin{bmatrix} 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & 0 & \boldsymbol{I} \end{bmatrix}. \tag{3}$$

For tracking control, the system will use an architecture of memoryless linear decentralized controllers to define control input $\boldsymbol{u}_i$ to each vehicle, similar to the description in Xue, et al. (2014)[5] and shown in Equation (4). We define a controlling matrix K which weighs the measurements of the vehicles in the swarm. The input for this matrix is the $\boldsymbol{y}$ component of the output vector of Equation (2), while the output is the linear combination of the position states of neighboring vehicles and the difference of the current location of the individual agents from the fixed-target tracking location.

$$\begin{bmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_n \end{bmatrix} = \begin{bmatrix} K_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & K_n \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_n \end{bmatrix} \tag{4}$$

3.2 Adversarial Model

As previously mentioned, we assume our adversary is able to make local measurements of the system dynamics over the time interval $\begin{bmatrix} 0 & t_f \end{bmatrix}$. The adversary is constrained to hacking into a single vehicle which, without loss of generality, we will assume is vehicle 1. Thus, the *adversary's measurements* become $\boldsymbol{y}_1$. Effective formation control schemes maintain observability among agents, which is precisely what creates the problem–by hacking into a one vehicle, an adversary can learn everything about the entire team with an effective estimator.

Aside from the adversary's measurements, we also assume that the attacker has the perspective of an "insider" or someone who is familiar with the system, possibly a rogue member of friendly forces. This means that the adversary has complete knowledge of the model of the UAV swarm, to include the identities of the vehicles being measured, the internal dynamics of the vehicles, and communication and sensing abilities of each. This perspective enables the framework to conduct insider-attack threat analysis, where an attack occurs through the channel of an individual or team with some measure of authorized access that is abused or used maliciously, and it provides a kind of "worst-case" analysis.

Assuming the attacker has access to the team model and control protocols ensures that as long as the formation controllers maintain observability among agents, an effective attacker can build the necessary estimators to learn all desired information about the entire team, including estimation of each vehicle's home base or the target location. This exposure of the state of the vehicle network could then lead to possible attacks on the home bases of each vehicle, the interception of the target of a search and rescue mission in hostile territory, or the destruction of the vehicles themselves, among other outcomes.

4. SECURING DRONE SWARMS

The key technology proposed here to secure the drone network is *privacy-preserving computation*, specifically secure multiparty computation (MPC)[6,7] due to its efficiency.[8] Homomorphic encryption may be an alternative depending on the specific needs for privacy-preserving computation. MPC offers a way for mutually distrusting parties to compute functions of private values without revealing the values. In MPC this can be achieved by computing secret shares of private input values[9] or by using garbled circuits.[10] For the purposes of our analysis framework, we view these techniques as ensuring that an attacker can only intercept a function of the previously available measurements. We do this to account for information leaked to the attacker that stem from the computations that are executed privately. We model this with a projection operator $h^T = \begin{bmatrix} h_1 & h_2 & \dots & h_p \end{bmatrix}$ which multiplies $\boldsymbol{y}_1$ to yield the hacker's measurement, $y_h = h^T \boldsymbol{y}_1$. This measurement is not necessarily secure, meaning that an attacker still might be able to estimate all state information about the swarm if the system is observable from y_h. Thus, the key is to engineer the privacy-preserving computation to ensure that critical state information is *not* observable from y_h. Doing so guarantees that even with insider information, the drone network is safe from state inference attacks. Thus, we use the analytical framework to discover the best method for applying privacy-preserving computation to the system.

To accomplish this, we design h to ensure that swarm observability is destroyed from y_h. This is done by choosing h so that $h^T C_1$ is orthogonal to at least one eigenvector of A. We have some design freedom about which eigenvector or eigenvectors we choose, so this enables one to protect the most critical modes of the system.

The ability of a swarm of UAVs to complete a collaborative mission is contingent on the condition that the swarm members are able to interact, through their communication and sensing abilities. However, this same condition, which allows a swarm to interact and work together, also implies that the swarm is vulnerable to observation or spying attacks. Having the ability to listen into a single drone gives the adversary the opportunity to infer the locations of the home bases of neighboring UAVs, current location or status in flight of the vehicles, or probable location of the intended mission objective. We illustrate how such an attack is made feasible by the system dynamics that make swarm observation possible.

4.1 Vulnerable Scenario

Following the DIN model described in the previous section as well as the scenario involving 4 UAVs, we envision these vehicles to be a coordinated swarm tasked with completing a search sweep for a search and rescue operation.

The UAVs are tasked with tracking to a fixed location. They are equipped with onboard cameras which scan for the target of this hypothetical search and rescue mission in a hostile environment.

Assume that the adversary is only able to compromise one of the drones in this team, which without loss of generality we will assume is drone 1. As discussed, the attacker is supposed to have full state measurement of the drone and that of neighboring drones as well with which the compromised unit is communication with or sensing. This is equivalent to reading the complete output vector $\boldsymbol{y}_i = C_i \boldsymbol{x}_i$ where the ith drone is compromised. In this example, the attacker's measurements consist of the position and velocity of drone 1 and it's sole neighbor drone 2.

On the surface, this appears to be an unfortunate, but not necessarily catastrophic scenario since only 2 of the 4 drones have been compromised. However under certain conditions this signifies the exposure of information about the entire network of UAVs. By definition of the Popov-Belevitch-Hautus (PBH) test[11] for Linear-Time-Invariant (LTI) system observability, the pair (A, C) is observable if and only if $\begin{bmatrix} sI - A & C \end{bmatrix}^T$ is full column rank for $s \in \mathbb{C}$. Unfortunately, this condition is common given the interactivity conditions that are required in order to have an environment where a UAV swarm can cooperate and complete a shared objective.

4.2 Fortifying the Swarm

The underlying vulnerability in our system of UAVs is the communication required to collaborate on the task at hand. The individual machines need to compute relative or absolute state changes in relation to the target objective or other units in the swarm and then actual controls based on those computations. This introduces the vulnerability, since by successfully compromising one or more drones, the adversary can then recreate the system state at any point in time. In order to secure the system, we need to design h^T such that $\begin{bmatrix} sI - A & h^T C_i \end{bmatrix}^T$ drops rank for a chosen $s \in \mathbb{C}$. This makes it so that the adversary is unable to estimate *all* state information. Nevertheless, the attacker may still be able to infer some other critical information about the system. Complete system security would be dependent upon forcing the condition $h^T C_i \boldsymbol{x} = 0$ for all eigenvalues x of A. This constraint forms a tradeoff between limiting the behavior of the system dynamics and the securing the entire system.

Actual implementation of securing the drone swarm is possible through the use of MPC.[6] The vehicle network can be completely or mostly secured to allow for the execution of collaborative missions. MPC preserves the privacy of the states of the other members of the swarm while allowing the necessary computations by the local vehicle's controller, thwarting the observational attack vector of our hypothetical adversary. The effect of MPC is to reduce the dimensionality of C_H such that C_H becomes orthogonal to the observable subspace of the system. This implies that the initial conditions become indistinguishable to the adversary, and she cannot infer some or all information about the current or past states of the system without other data.

4.3 Simulation

We demonstrate the feasibility of an observation attack on the UAV network described in Figure 1 by means of a simulation on the framework described in Section 3. Under the assumptions made previously about our adversary and the framework, we simulated the dynamics of the system with a static, stabilizing controller. The individual agents complete a tracking assignment, starting from their respective initial conditions to either a relative location within the network or to an absolute location, depending on if the agent is a leader or follower within the system.

Recall the abstract information graph from Figure 1b, noting that if the attacker has access to the first agent, then she has the ability to observe direct measurements from the first and second drones. In this scenario, the fourth agent has been designated the leader and given an absolute location to converge on, while the other agents are given relative locations to the next drone to follow. In this simulation, the fourth drone is assigned to track to position 0, and drones 1 - 3 are assigned to track at distances -20, -10, and 5 from drones 2 - 4, respectively.

In the top left panel of Figure 2, we see the actual positions of each UAV as they track from their starting locations to the final objective. The top right panel shows the estimation of the system states from the adversary in the vulnerable scenario while the bottom left is with the secured system. Note that the adversary is largely able to estimate the positions of each UAV, despite only having compromised one vehicle in the vulnerable

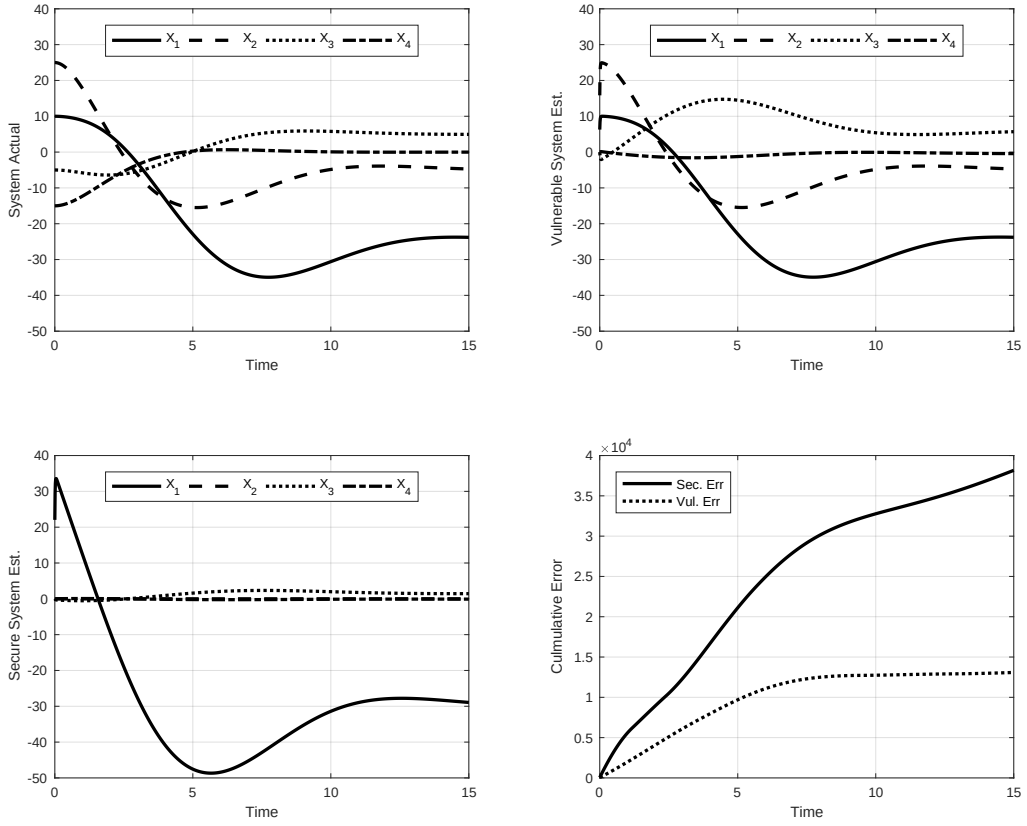


Figure 2: Simulation of previous formation with vulnerable and secure configurations

scenario. On the other hand, with the secured system in place, the adversary is only able to effectively track the compromised drone. The final, bottom right panel shows the cumulative error across all vehicles over the course of the simulation. While the vulnerable system estimate converges to the actual system states, the secured system estimates never effectively converge for the non-compromised agents.

5. CONCLUSION

As technology and governing policies continue to encourage the use of networks of UAVs, system security will only continue to increase in importance. Information breaches are only becoming more damaging as sensitive and critical processed come to rely on increasingly brittle infrastructure. Tools like vulnerability analysis and multiparty computation will become best-practices in order to utilize technical advantages while concurrently not exposing unknown weaknesses. We have shown here how a simple network of UAVs tasked in a search and rescue operation were vulnerable to observational attacks from a malicious adversary. Through the use of MPC, this threat can be mitigate to an acceptable level dictated by mission objectives and resource constraints.

Future work will include continuing with this methodology to understand which links of a given network are the most vulnerable or have the greatest value to an adversary. This poses the question of identifying an automated process to highlight these particular links, prioritizing them by most to least vulnerable. Then

cost analysis of securing the system can be conducted to understand how much effort would be requiring in securing a particular system to an acceptable level. We envision the framework to provide critical information in formulating a system protection strategy and as being useful in all stages in a system development lifecycle. Ideally the framework will be used during early development planning stages, but is still extremely useful after a system has already been built and deployed and is in a maintenance stage. This gives system owners and operators the ability to explore weaknesses in their system and choose both optimal protections and protection strategies.

REFERENCES

[1] Dubins, L. E., "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics* **79**(3), 497–516 (1957).

[2] Owen, M., Beard, R. W., and McLain, T. W., "Implementing dubins airplane paths on fixed-wing uavs," in [*Handbook of Unmanned Aerial Vehicles*], 1677–1701, Springer (2015).

[3] Chapman, A. and Mesbahi, M., "Uav swarms: models and effective interfaces," in [*Handbook of Unmanned Aerial Vehicles*], 1987–2019, Springer (2015).

[4] Roy, S., Saberi, A., and Herlugson, K., "Formation and alignment of distributed sensing agents with double-integrator dynamics and actuator saturation," *Sensor Network Applications* , 1–50 (2004).

[5] Xue, M., Wang, W., and Roy, S., "Security concepts for the dynamics of autonomous vehicle networks," *Automatica* **50**(3), 852 – 857 (2014).

[6] Clark, M. R. and Hopkinson, K. M., "Transferable multiparty computation with applications to the smart grid," *IEEE Transactions on Information Forensics and Security* **9**(9), 1356–1366 (2014).

[7] Clark, M. R., Stewart, K., and Hopkinson, K. M., "Dynamic, privacy-preserving decentralized reputation systems," *IEEE Transactions on Mobile Computing* **16**(9), 2506–2517 (2017).

[8] Clark, M. R. and Hopkinson, K. M., "Towards an understanding of the tradeoffs in adversary models of smart grid privacy protocols," in [*Power and Energy Society General Meeting (PES), 2013 IEEE*], 1–5, IEEE (2013).

[9] Shamir, A., "How to share a secret," *Communications of the ACM* **22**(11), 612–613 (1979).

[10] Yao, A. C., "Protocols for secure computations," in [*Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*], 160–164, IEEE (1982).

[11] Hespanha, J. P., [*Linear systems theory*], Princeton university press (2018).