Vulnerability Analysis of Infrastructure Systems

Sean Theodore Lane
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Physical Sciences and Mathematics Commons

BYU ScholarsArchive Citation
Lane, Sean Theodore, "Vulnerability Analysis of Infrastructure Systems" (2020). Theses and Dissertations. 8614.
https://scholarsarchive.byu.edu/etd/8614

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
Vulnerability Analysis of Infrastructure Systems

Sean Theodore Lane

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Sean Warnick, Chair
Casey Deccio
Eric Mercer

Department of Computer Science
Brigham Young University

Copyright © 2020 Sean Theodore Lane
All Rights Reserved
ABSTRACT

Vulnerability Analysis of Infrastructure Systems

Sean Theodore Lane
Department of Computer Science, BYU
Master of Science

Complex cyber-physical systems have become fundamental to modern society by effectively providing critical services and improving efficiency in various domains. Unfortunately, as systems become more connected and more complex, they also can become more vulnerable and less robust. As a result, various failure modes become more common and easily triggered from both unanticipated and malicious perturbations.

Research has been conducted in the area of vulnerability analysis for cyber-physical systems, to assist in locating these possible vulnerabilities before they can fail. I present two case studies on different forms of critical infrastructure systems to identify vulnerabilities and understand how external perturbations can affect them, namely UAV drone swarms and municipal water infrastructure. Specifically, this work:

1. Illustrates a spying attack on an unmanned aerial vehicle swarm performing a collaborative task, along with how the problem could be mitigated by using secure multiparty computation,

2. Presents an attack to disrupt the service of a hypothetical municipal water system, through the control of speed settings on booster pumps,

3. Demonstrates the use of system models to highlight inherent system vulnerabilities.

Keywords: cyber-physical system, vulnerability, critical infrastructure, UAV swarm, municipal water
ACKNOWLEDGMENTS

I would like to give special thanks to my wife, Ashley, for her eternal love and enduring support throughout my graduate education, as well as her assistance in improving the clarity of this thesis. Her persistent encouragement and steadfast presence mean more than she will ever know. I would also like to thank my advisor, Dr. Sean Warnick, for supporting me in this work and my educational journey, and who has helped me become the person I am today. I am grateful for the assistance and guidance from my colleagues of the BYU IDeA Labs. Thanks also to my committee members for their guidance, feedback, and time given throughout this process. Lastly, I would like to express my gratitude to God for the many blessings and opportunities that we have received, and for His love that we have felt in our time at this institution.
Table of Contents

List of Figures vi

1 Introduction 1
   1.1 Cyber-physical systems ........................................... 1
   1.2 Motivation ......................................................... 2
   1.3 Contributions .................................................... 2

2 Related Work 4
   2.1 Empirical studies .................................................. 4
   2.2 Agent-based models ............................................... 5
   2.3 Graph theoretic and network topology approaches ............... 5
   2.4 System dynamics .................................................. 6

3 UAV Swarms 7
   3.1 Threat Model ...................................................... 7
   3.2 System Modeling Framework ....................................... 8
      3.2.1 Double-Integrator-Network Model ............................ 9
      3.2.2 Adversarial Model ........................................... 11
   3.3 Secure Multiparty Computation ................................... 12
   3.4 Vulnerable Scenario ............................................... 13
   3.5 Fortifying the Swarm .............................................. 14
   3.6 UAV Swarm Simulation ............................................ 15
4 Municipal Water Systems

4.1 Variable Frequency Drive .................................................. 18
4.2 Induction Motor ............................................................. 20
4.3 Hydraulic Network .......................................................... 22
4.4 Water System Simulation ................................................. 23

5 Conclusion & Future Work .................................................. 28

References ........................................................................... 30
List of Figures

3.1 Example UAV Team Configuration .............................................. 10

3.2 Simulation of previous formation with vulnerable and secure configurations.
The top left panel illustrates the ground truth values of the state value, or
location, of each agent over the course of the simulation. The top right is
the estimate of these state values in the vulnerable scenario, which we can
see closely tracks the actual values. The bottom right panel shows the same
estimates but in the secure scenario where the system observability is reduced.
This shows that estimates never converge to the true system values, rendering
the adversary unable to estimate the agent positions. The final bottom right
panel further illustrates this point, showing how the cumulative estimate errors
reach an upper limit in the vulnerable scenario while the cumulative error in
the secure scenario fails to converge. ........................................... 16

4.1 Images showing the relations between the three components that link power
and water distribution networks: the variable frequency drive, the induction
motor, and the centrifugal pump. ............................................. 18

4.2 Simulation of the induction motor behavior, showing internal motor state
values and rotor velocity. ..................................................... 24

4.3 Simulation of the water system under nominal conditions ................. 26

4.4 Simulation of the water system under attack conditions ................ 27
Chapter 1

Introduction

The intersection between cyber and physical systems has grown an inordinate amount as digital extensions to existing systems and processes becomes increasingly common. This naturally has grown to include critical infrastructure systems on which entire nations and societies place their foundations to provide efficient and effective services to citizens and shareholders alike [35]. The goal of this thesis will be to use existing work on vulnerability analysis of such systems and show its applicability to other, different domains, namely UAV drone swarms and water distribution infrastructure. This section will provide further background and highlight the motivation for why it should be addressed.

1.1 Cyber-physical systems

Cyber-physical systems (CPS) are “physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core” [45]. Most developed nations have a vast amount of infrastructure implemented as CPS, systems which are “…needed for the functioning of a community or society” [35]. Examples of such types of systems include power and water distribution and generation (both at regional and municipal levels), emergency services and their communications platforms, transportation networks, Internet and consumer communications networks, banking and finance systems, public health services, fuel and oil production and storage, etc. The scope of this paper will focus on the application of vulnerability research to produce case studies of three different
forms of infrastructure systems: unmanned aerial vehicle (UAV) drone swarms and municipal power & water distribution systems.

1.2 Motivation

One hardly needs to search news articles for problems relating to cyber physical system failures and security breaches. Examples of attacks include industrial systems [20], transportation networks [25], power generation [42], and more. Power and water networks are obvious examples, in that each directly or indirectly provide for basic human needs like hydration, food preparation, and warmth. UAV swarms are arguably the least likely choice among the three case studies in critical infrastructure systems, but are rapidly being deployed in a number of environments to accomplish various objectives and are considered to be a likely tool in the future arsenal of military forces, police agencies, firefighting departments, and others [1, 26, 30, 36].

As cyber-physical systems become more common and prevalent throughout the world, security and robustness of these systems becomes an increasingly important consideration. Security breaches that were once relatively isolated inconveniences for a region or even nation at large become catastrophic points of failure that can cause millions of dollars in damage [3] and even loss of life [31, 33]. Securing these systems and ensuring their robust automation is a priority of immediate concern in order to mitigate these potential failures before they can occur.

1.3 Contributions

This thesis makes three direct contributions. First, it demonstrates a spying or observation attack on a cyber physical system in the form of an unmanned aerial vehicle swarm. The context of this scenario is that the attacker makes use of insider knowledge of the swarm dynamics, but through the application of secure multiparty computation to reduce the observability of the system, this attack can be thwarted [14, 24]. A software implementation
of the system model was created and then simulated under both vulnerable and secured scenarios to show the effects of estimating the system state within each scenario. Second, it presents an attack designed to deny the service of a municipal water system by maliciously adjusting the speed set points of the booster pumps used to maintain pressure or head-level within the network. The novelty of this portion comes from the implementation of defined models for the individual components comprising the booster pumps from existing academic work [17, 29, 40]. These models were then validated and integrated with existing state-of-the-art hydraulic modeling software. They are then used to define the behavior of the pumps when simulating the water network, showing how effects can propagate across the entire system from key components. Lastly, this work illustrates the use of system models to highlight inherent system vulnerabilities that can be obscured by the complexity of the system itself, and provides a foundation for future extensions of vulnerability analysis with these types of infrastructure.
Chapter 2

Related Work

A literature review in this area reveals that a wide variety of different approaches have been taken in analyzing infrastructure vulnerabilities. As noted in [55], these can be largely delineated into empirical and predictive approaches, with the former attempting to garner insight and intuition from past failures, and the latter to attempt to find inherent weaknesses before they have the opportunity to strike. The predictive methods include agent-based models, modeling based on economic goals, graph theoretic and network topology approaches, and those utilizing system dynamics. In this section, we will focus on exploring works related to modeling infrastructure and cyber-physical systems and their related approaches.

2.1 Empirical studies

Utne et al. [54] present a framework to evaluate the cross-connections of critical infrastructure systems as part of a comprehensive analysis of system vulnerability. This framework is then used as part of a case study in a partnership with the Emergency Preparedness Group and the city of Oslo, Norway. A more infamous example includes case studies of the 2000 Maroochy Shire water systems breach in Australia. Abrams and Weiss [3] describe in detail the events and circumstances leading up to the failure and the events that lead to the enablement of the attack. While certainly valuable as a teaching tool and for gaining intuition behind avenues of potential failures, empirical analysis can only be performed on system failures after a failure has occurred. This weakness is what incentivizes researchers to identify predictive methods to find failure modes before they are triggered.
2.2 Agent-based models

Agent-based models start with the assumption that from many simple interactions that occur between relatively simple agents, complex behavior and patterns emerge, as opposed to the complexity of a few components themselves. By giving software-implemented agents simple rules with which to react to their immediate environment, instead of global goals for the entire system, the agents act simply for their own benefit but in aggregate exhibit complex behavior expected from the system as a whole. Examples of agent-based models for vulnerability analysis can be found in Barton et al. [6] and Oliva et al. [38]. However, these models are often focused on the economic impact of one infrastructure layer on another across a multitude of systems, as opposed to physical damage that this work seeks to measure and mitigate. Furthermore, it is often difficult to validate the analytical analysis of these types of systems due to the difficulty in obtaining data of the appropriate granularity for the systems that are being modeled.

2.3 Graph theoretic and network topology approaches

Another approach to modeling system vulnerabilities stems from graph theory and analyzing the topology of the networks involved. Havlin et al. [23] presented a graph theoretic approach to modeling interdependent systems and then analyzed the case of an Italian national electrical outage in September 2003. Their work showed the critical number of nodes of the interconnected graphs that would lead to a complete fragmentation of the underlying networks, modeled in the cited paper as two Erdös-Rényi networks. The benefit of this approach is the ability to see the higher level effects of node shutdown as the failures cascade throughout the system. However, the lack of the incorporation of physical dynamics reduces the accuracy of this approach when considering cyber-physical infrastructure systems.
2.4 System dynamics

An example of system vulnerability analysis that incorporates system dynamics was produced by Yeung et al. [58]. The gold standard for water distribution simulation is the EPANET water distribution network model [46], which is used broadly among commercial utility companies as well as government agencies. This network utilizes conservation equations that satisfy conservation laws of mass, nodal flows, and boundary conditions among demand and source nodes (such as reservoirs, rivers, tanks, etc). These are discussed in further depth in Chapter 4 where this system is used in analyzing the affects of attacking pump settings within the network. Further examples of this work include analysis of the Sevier River system found in central Utah as conducted by [22, 34].
Chapter 3

UAV Swarms

This chapter demonstrates a case study of modeling and assessing the integrated cyber-physical dynamics of unmanned aerial vehicle (UAV) swarms through a search-and-rescue (SAR) example, highlighting its benefits even in the face of “insider” attackers who have prior understanding and knowledge of the system structure. It uses the results of a vulnerability 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. This 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 of the allocation of limited defensive resources and results in more efficient investment in the cybersecurity of complex systems.

3.1 Threat Model

Cybersecurity implementations often rely on existing security infrastructure, protocols, and practices to maintain system integrity. However, due to expense and difficulty, security is often 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 which are cooperating in a SAR mission over a predefined track or pattern.
The attack being modeled in this scenario will be an observation, inference, or spying attack 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 SAR 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 of the UAVs and can actively listen to the information being sent and received by the compromised UAV. She understands the objective of the swarm in as much as the swarm is participating in a SAR operation, and she intends to utilize this information in an adversarial manner to the detriment of the swarm or friendly forces.

3.2 System 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. The literature shows a number of peer-reviewed methods of modeling UAVs cooperating in a shared task. These include using Dubins path generation techniques [19] on fixed-wing UAVs [39], consensus dynamics within adjacency graphs [9], and double-integrator-network (DIN) models based on physical first principles [47]. Our model is based on the previous research by Xue, et al. (2014)[56], 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. We now review the DIN and adversary models used in this case example.
3.2.1 Double-Integrator-Network Model

We will consider a team of \( n \) UAVs labeled as \( i = 1 \ldots n \), 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 \( D = 0 \) for the remainder of this paper.

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

\[
\dot{x}_i = A_i x_i + B_i u_i \quad y_i = C_i x_i \tag{3.1}
\]

where \( u_i, x_i, y_i \) 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

\[
\begin{bmatrix}
\dot{x} \\
y
\end{bmatrix} =
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}
\begin{bmatrix}
x \\
u
\end{bmatrix}
\triangleq
\begin{bmatrix}
\dot{x}_1 \\
\vdots \\
\dot{x}_n \\
y_1 \\
\vdots \\
y_n
\end{bmatrix} =
\begin{bmatrix}
A_1 & 0 & B_1 & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & A_n & 0 & B_n \\
C_1 & 0 & \ddots & 0 \\
\vdots & \ddots & \ddots & 0 \\
0 & C_n
\end{bmatrix}
\begin{bmatrix}
x_1 \\
\vdots \\
x_n \\
u_1 \\
\vdots \\
u_n
\end{bmatrix} \tag{3.2}
\]

Another assumption we make is that the UAVs must communicate and coordinate 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 3.1 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 SAR.

(b) Abstract information graph corresponds to the formation depicted in Figure
3.1a. Note that the analysis techniques used in this work allow for arbitrary
team formations; this simple structure is used here for pedagogical clarity.

Figure 3.1: Example UAV Team Configuration

In this scenario, the $C$ matrix of the total system is set so that drones not only measure
their own 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

$$C_1 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \quad C_2 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \quad C_3 = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & I \end{bmatrix}, \quad C_4 = \begin{bmatrix} 0 & 0 & I \\ 0 & 0 & I \\ 0 & 0 & I \end{bmatrix}.$$ (3.3)
For tracking control, the system will use an architecture of memory-less linear decentralized controllers to define control input $u_i$ to each vehicle, similar to the description in Xue, et al. (2014)[56] and shown in Equation (3.4). We define a controlling matrix $K$ which weighs the measurements of the vehicles in the swarm. The input for this matrix is the $y$ component of the output vector of Equation (3.2), 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.

$$
\begin{bmatrix}
    u_1 \\
    \vdots \\
    u_n
\end{bmatrix}
=
\begin{bmatrix}
    K_1 & 0 \\
    \ddots & \ddots \\
    0 & K_n
\end{bmatrix}
\begin{bmatrix}
    y_1 \\
    \vdots \\
    y_n
\end{bmatrix}
$$

$$
(3.4)
$$

### 3.2.2 Adversarial Model

As previously mentioned, we assume our adversary can make local measurements of the system dynamics over the time interval $[0, t_f]$. 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 $y_1$. Effective formation control schemes maintain observability among agents, which is precisely what creates the problem—by hacking into 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 vehicle. 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 SAR mission in hostile territory, or the destruction of the vehicles themselves, among other outcomes.

3.3 Secure Multiparty Computation

The key technology proposed here to secure the drone network is privacy-preserving computation, specifically secure multiparty computation (MPC) [13, 14] due to its efficiency [12]. 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 [48] or by using garbled circuits [57]. 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 $h^T = \begin{bmatrix} h_1 & h_2 & \ldots & h_p \end{bmatrix}$ which multiplies $y_1$ to yield the hacker’s measurement, $y_h = h^T y_1$. 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 can 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 the 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.

### 3.4 Vulnerable Scenario

Simulating the UAV swarm under the conditions and assumptions that were previously described, we can now see how the entire team becomes vulnerable to observation. 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 communicating with or sensing. This is equivalent to reading the complete output vector $y_i = C_i x_i$ where the $i$th drone is compromised. In this example, the attacker’s measurements consist of the position and velocity of drone 1 and its sole neighbor drone 2.

On the surface, this appears to be an unfortunate but not necessarily catastrophic scenario since only information from 2 of the 4 drones has 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 [24] 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 to have an environment where a UAV swarm can cooperate and complete a shared objective.

3.5 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 actuate 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

$$sI - A \cdot h^T C_i$$

drops rank for a chosen $s \in \mathbb{C}$. This makes it so 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 $h^T C_i \cdot x = 0$ for all eigenvalues $x$ of $A$. This constraint forms a trade-off between limiting the behavior of the system dynamics and securing the entire system.

The actual implementation of securing the drone swarm is possible through the use of MPC [13]. The vehicle network can be either 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.
3.6 UAV Swarm Simulation

We demonstrate the feasibility of an observation attack on the UAV network described in Figure 3.1 using a simulation on the framework described in Section 3.2. 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 3.1b, noting that if the adversary has access to the first agent, then she can 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 our 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 3.2, we see the actual positions of each UAV as they track from their starting locations to the final objective. These are the state values that serve as the ground truth which the adversary wants to estimate. The top right panel shows the estimation of the system states from the adversary in the vulnerable scenario. Initially, the estimates of the agent locations have a large amount of error but they quickly converge to the actual values. This indicates that the adversary is largely able to estimate the positions of each UAV, despite only having compromised one vehicle in the vulnerable scenario. Contrast this with the bottom right panel which shows the adversarial estimates in the secure scenario with reduced system observability. The subfigure shows that aside from the compromised agent, the estimates of the system states never converge to the true values. Thus the adversary is only able to track the compromised drone. The final, bottom right panel shows the cumulative error across all vehicles over the course of the simulation. It shows that the cumulative error of the adversarial estimates begin to plateau towards an
Figure 3.2: Simulation of previous formation with vulnerable and secure configurations. The top left panel illustrates the ground truth values of the state value, or location, of each agent over the course of the simulation. The top right is the estimate of these state values in the vulnerable scenario, which we can see closely tracks the actual values. The bottom right panel shows the same estimates but in the secure scenario where the system observability is reduced. This shows that estimates never converge to the true system values, rendering the adversary unable to estimate the agent positions. The final bottom right panel further illustrates this point, showing how the cumulative estimate errors reach an upper limit in the vulnerable scenario while the cumulative error in the secure scenario fails to converge.

upper limit within the vulnerable scenario while the cumulative error of the estimates in the secure scenario continuously grow. This indicates that by reducing system observability, the adversary is unable to accurately estimate the system states of the UAV drone swarm to ascertain the location of the uncompromised agents.
Chapter 4

Municipal Water Systems

When considering fundamental infrastructures that support basic necessities, an obvious example is water distribution. This chapter demonstrates a case study of modeling a denial-of-service attack on a hypothetical municipal water system which is done by manipulation of the speed control of induction motors powering booster pumps. These pumps maintain adequate head level, a measure of pressure commonly used in practice, across the system. It makes use of a detailed model of the induction motor and variable frequency drive system to determine the speed behavior of the booster pumps, which is then fed into the hydraulic network simulation.

While pressure is introduced into the water distribution system from source nodes with fixed head levels [49], like reservoirs and rivers, booster pumps are required to maintain adequate amounts of pressure farther away from these source nodes. The booster pumps are composed of three principle subcomponents, which are illustrated in Figure 4.1:

1. Variable frequency drive (VFD): Responsible for consuming three-phase, alternating current power which is rectified to direct current and then uses an inverter to transform the direct current back into three-phase, alternating current with some modification to control the induction motor power output, as shown in Figure 4.1a,

2. Induction motor: The motor, shown in Figures 4.1a and 4.1b, consumes modified power from the VFD and uses magnetic induction to efficiently produce angular torque in the form of a spinning rotor,

17
3. Centrifugal pump: As shown in Figure 4.1b, the centrifugal pump is attached to the rotor of the induction motor and exerts pressure on the water distribution system downstream from the source nodes, maintaining head-level to meet predicted demands on the system farther away.

This approach of modeling the pump systems along with the hydraulic network provides a faithful simulation of the dynamics that are found within a hydraulic network under a denial-of-service attack and can provide the foundation for future work on different forms of attacks on municipal water networks. This should reduce costs associated with protecting such systems because it allows for the prioritization of the allocation of limited defensive resources and results in more efficient investment in cyber security.

4.1 Variable Frequency Drive

A standard water booster pump is outfitted with a VFD which is used to control the speed of the induction motor powering a centrifugal pump, thereby controlling the output pressure. The VFD itself is composed of three subcomponents: a rectifier, a direct current (DC) power link, and a power inverter. The rectifier is responsible for receiving three-phase alternating
current (AC) power and transforming it into DC power, which is the input into the DC link. The DC link is composed of a capacitor and an R-L filter, which smooths the pulses on the DC link that propagate from the rectifier. Finally, the DC power from the second VFD subcomponent is converted back into controlled three-phase AC power by means of the inverter, which is implemented by a pulse-width modulation (PWM) converter. Further details and diagram representations of the VFD subcomponents, as well as those of the induction motor, are found in [17, 29], and Equations 4.1–4.11 are the differential algebraic equations which describe the dynamics of the VFD.

\[ V_{d0} = \frac{3\sqrt{2}}{\pi} \cdot V_s \]  \hspace{1cm} (4.1)
\[ \frac{d(\Delta \omega)}{dt} = -\frac{K_P(\omega_{\text{ref}} - \omega_m)}{T_m(S)} + K_I(\omega_{\text{ref}} - \omega_m) \]  \hspace{1cm} (4.7)
\[ I_i = \frac{\sqrt{3}}{2} m V_{dc} \]  \hspace{1cm} (4.2)
\[ \frac{dV_{dc}}{dt} = \frac{1}{C}(I_r - I_i) \]  \hspace{1cm} (4.3)
\[ \frac{d(\omega_m)}{dt} = \frac{\omega_m' - \omega_m}{T_m(S)} \]  \hspace{1cm} (4.8)
\[ V_m = \frac{V_{dc}\sqrt{3}}{2\sqrt{2}} \]  \hspace{1cm} (4.4)
\[ v_a = \sqrt{2}V_m \cos(\omega_m t) \]  \hspace{1cm} (4.9)
\[ v_b = \sqrt{2}V_m \cos(\omega_m t - 2/3\pi) \]  \hspace{1cm} (4.10)
\[ v_c = \sqrt{2}V_m \cos(\omega_m t + 2/3\pi) \]  \hspace{1cm} (4.11)

\[ \frac{dS}{dt} = (T_e - T_m(S))/2H \]  \hspace{1cm} (4.5)
\[ \frac{dL_r}{dt} = \frac{1}{L} \cdot [V_{a0} - I_r(R + R_s) - V_{dc}] \]  \hspace{1cm} (4.6)

From above, \( T'_0, X_0, X' \) are motor parameters as defined by [17]. \( S \) is the slip between the rotor and stator angular velocities, and \( H \) is the inertia constant. \( L_r, L_s, R_r, R_s \) refer to
the inductance and the resistance of the rotor and stator, respectively, while \( C, L, \) and \( R \) refer to the capacitance, inductance, and resistance of the DC link. \( V_{d0} \) is the input voltage into the rectifier and \( V_d \) is the output, while \( V_s \) is the root mean square (RMS) voltage from the power grid and \( V_m \) is the RMS voltage output to the motor. \( I_i \) describes the current of the inverter, \( I_r \) is the current of the rectifier, and \( V_{dc} \) is the voltage at the capacitors which help maintain even power when rectifying AC power to DC. Lastly \( v_a, v_b, \) and \( v_c \) are input voltages given to the induction motor. All of these constants, parameters, and variables are elements of \( \mathbb{R} \).

### 4.2 Induction Motor

The second main component of the booster pumps is the induction motor along with the centrifugal pump that transforms electrical power into increased water pressure. Most industrial pumps utilize three-phase AC induction motors due to smoother and more balanced operation, in addition to usually being more economical for larger, fixed installations which are commonly used for booster pumps. The dynamics of the induction motor are described in Equations 4.12–4.24.

\[
\Psi_{mq} = X_{mq} \left( \frac{\Psi_{qs}}{X_{ls}} + \frac{\Psi_{qr}}{X_{lr}} \right) \tag{4.12}
\]

\[
\Psi_{md} = X_{md} \left( \frac{\Psi_{ds}}{X_{ls}} + \frac{\Psi_{dr}}{X_{lr}} \right) \tag{4.13}
\]

\[
i_{qs} = \frac{1}{X_{ls}}(\Psi_{qs} - \Psi_{mq}) \tag{4.14}
\]

\[
i_{ds} = \frac{1}{X_{ls}}(\Psi_{ds} - \Psi_{md}) \tag{4.15}
\]
\[ i_{qr} = \frac{1}{X_{lr}}(\Psi_{qr} - \Psi_{mq}) \]  
(4.16)

\[ i_{dr} = \frac{1}{X_{lr}}(\Psi_{dr} - \Psi_{md}) \]  
(4.17)

\[
\frac{d\Psi_{qs}}{dt} = \omega_b \left[ v_{qs} + \frac{R_s}{X_{ls}} (\Psi_{mq} - \Psi_{qs}) \right]
\]  
(4.18)

\[
\frac{d\Psi_{ds}}{dt} = \omega_b \left[ v_{ds} + \frac{R_s}{X_{ls}} (\Psi_{md} - \Psi_{ds}) \right]
\]  
(4.19)

\[
\frac{d\Psi_{qr}}{dt} = \omega_b \left[ v_{qr} + \frac{\omega_r}{\omega_b} \Psi_{dr} + \frac{R_r}{X_{lr}} (\Psi_{mq} - \Psi_{qr}) \right]
\]  
(4.20)

\[
\frac{d\Psi_{dr}}{dt} = \omega_b \left[ v_{dr} - \frac{\omega_r}{\omega_b} \Psi_{qr} + \frac{R_r}{X_{lr}} (\Psi_{md} - \Psi_{dr}) \right]
\]  
(4.21)

\[ X_{md} = X_{mq} = 1 / \left( \frac{1}{X_m} + \frac{1}{X_{ls}} + \frac{1}{X_{lr}} \right) \]  
(4.22)

\[ T_e = \frac{3}{2} \cdot P \cdot \frac{1}{\omega_b} \cdot (\Psi_{qr} i_{dr} - \Psi_{dr} i_{qr}) \]  
(4.23)

\[
\frac{d\omega_r}{dt} = \frac{P}{2J} (T_e - T_{load})
\]  
(4.24)

As with the VFD model, the following are all elements of \( \mathbb{R} \). The constants \( X_m, X_{ls}, \) and \( X_{lr} \) are defined motor parameters for the mutual, stator, and rotor inductance. \( P, J, \omega_b \) are constants for the number of poles of the motor, the motor inertia constant, and the frequency of the power input passed into the motor. \( T_e \) is the electrical torque generated, \( T_{load} \) is the torque exerted on the rotor in reaction to the electrical torque. The variables
\(v_{dr}, v_{ds}, v_{qr}, v_{qs}\) are the voltages of the rotor and stator in the synchronous rotating reference frame after transforming the input voltages with the DQ0 or Park Transform as described by Park [41]. The variables \(i_{dr}, i_{ds}, i_{qr}, i_{qs}\) are similar but correspond to current, and \(\Psi_{dr}, \Psi_{ds}, \Psi_{qr}, \Psi_{qs}\) correspond to magnetic fluxes. Lastly, \(\omega_r\) is the output speed of the rotor.

4.3 Hydraulic Network

We turn to EPANET for simulating the water network system itself, which is industry-standard software for modeling the hydraulic and water quality behavior of distribution pipe networks [46]. The solver implemented in this software satisfies various constraints in the form of conservation laws in mass, source and demand boundary conditions, and nodal inflows and outflows. These constraints can be mathematically formulated as

\[
A_{12}H = F(Q, r, n_e) - A_{10}H_0 \tag{4.25}
\]

\[
A_{21}Q = q
\]

with \(H \in \mathbb{R}^n, Q \in \mathbb{R}^{np}, r \in \mathbb{R}^{np}, H_0 \in \mathbb{R}^{ns},\) and \(q \in \mathbb{R}^{nd}\). Each element of \(H\) specifies the head level, or pressure, at each of the \(n\) nodes within the network, which can vary throughout the simulation. \(Q\) is a vector containing the flow rate in each of the \(np\) pipes connecting the nodes. The vector \(r\) specifies the resistance coefficients satisfying one of the head-loss formulas that EPANET has implemented: Hazen-Williams, Darcy-Weisbach, or Chezy-Manning [46]. These models describe how head or pressure is lost due to friction with the pipe walls. Hazen-Williams is the most commonly used formula within the United States, but it is limited to systems carrying water and that have turbulent flows. Darcy-Weisbach doesn’t have these limitations, but is also more computationally expensive, and Chezy-Manning is generally used for open channel flow simulation. This simulation will use the Hazen-Williams formula which is standard for municipal water systems. The vector \(H_0\) defines the head levels for the
fixed-head sources, like reservoirs, rivers, tanks, etc, while \( q \) is a vector where each element is the demand at \( n_d \leq n \) nodes within the network. The matrix \( A_{12} \) is defined as

\[
[A_{12}]_{ij} = \begin{cases} 
1 & \text{if fluid from pipe } i \text{ enters node } j \\
0 & \text{if pipe } i \text{ and node } j \text{ are not connected} \\
-1 & \text{if fluid from pipe } i \text{ leaves node } j
\end{cases}
\]

while \( A_{10} \) is similarly defined for fixed-head source nodes, and \( A_{21} = A_{12}^T \). The function \( F : \mathbb{R}^{n_p} \to \mathbb{R}^n \) specifies the head-loss of the network as a function of the flow rates defined in \( Q \). The constant \( n_e \) is an empirically measured constant which is defined as 1.852 in Table 3.1 in [46].

### 4.4 Water System Simulation

Utilizing the models outlined in the previous sections, we can now carry out simulations of the individual components as well as the integrated system. The ultimate objective of this section will be show how compromising booster pump speeds affect water availability throughout the distribution system. Initial simulations of the isolated induction motor model without the VFD model and with a static load, along with the combined VFD and induction model along with a dynamic load demonstrate the anticipated behavior of the individual components. The motor begins from a standing start and Figure 4.2 shows the speed of the rotor converging on the reference point while the amount of magnetic flux in the rotor and stator of the motor oscillates from a static reference frame, as described by Krause and Thomas [29].

The next step is simulating the performance of booster pumps within a municipal water network and using that behavior as inputs into the simulation of the water network. The water network test case that will be used is a modified version of the EPANET Network 3 [46], which has 92 nodes, 114 links, 2 system sources, and utilizes 2 pumps. The network
Figure 4.2: Simulation of the induction motor behavior, showing internal motor state values and rotor velocity.

layout can be seen in Figure 4.3, and a majority of the nodes experience either constant or recurring demands with a 24-hour cycle to mimic the demands of a population on a water system. The pump speeds are passed into the system as input, illustrating the effects of manipulating the pumps on the head level of the water network. We then run two simulations, the first where the pump speeds remain fixed and run constantly, ensuring the head level of the entire network at a suitable level, and the second with altered pump speeds. Both simulations occur over a period of 7 days, with a simulated denial-of-service attack reducing pump speed first to 50%, then to 25%, and finally back to 50% for the duration of the simulation.

Figure 4.3 shows the system over this 7-day period under nominal conditions, with the head level used as the key indicator in these figures. The system performs well, consistently providing head levels of around 40 feet even during higher loads in the system. The red, vertical dashed line in Figure 4.3a indicates the time point of 48 hours, at which time the head level status across the entire network is shown in Figure 4.3b. The individual lines within 4.3a show the head level status throughout the simulation for a sampling of nodes.
found at various extreme edges of the network. These show the consistent water availability for the entire duration of the simulation, despite varying demands.

Similarly, Figure 4.4 shows the system over this same period but under the attack conditions previously described. Again, the red, vertical dashed line in Figure 4.4a indicates the time point of 48 hours, at which time the head level status across the entire network is shown in Figure 4.4b. The individual lines within 4.4a represent the head level status throughout the simulation for the same sampling of nodes as before, which found at various extreme edges of the network. These show the critical drop in water availability after the commencement of the attack, which continues for the entire duration of the simulation, along with the same water demands as before. The system performs poorly, with consistently low head levels which peak around 20 feet, and repeatedly hit zero during higher loads in the system.

After running initial validation simulations with the individual booster pump model, we then performed simulations on an integrated water simulation that incorporated the booster pump model under two different scenarios. The first scenario showed nominal performance under normal conditions, with booster pumps being utilized to meet regular demand constraints and adequately maintaining head levels throughout the system, indicating that water is consistently available. We contrast this with the results of the second scenario, which shows how the compromised booster pumps fail to adequately maintain head levels in various parts of the system. This would lead to conditions where water is unavailable from the distribution system, which could be catastrophic for a normal municipality.
(a) Head level states of various nodes under nominal conditions. The dashed red line indicates the time point at which the network in Figure 4.3b was created. The data in the graph corresponds to four nodes at separate points in the network, giving an overview of water availability throughout the duration of the simulation.

(b) Head level of the entire system after 48 hours under nominal conditions, showing that all nodes have similar head-levels, which is an indication of water availability.

Figure 4.3: Simulation of the water system under nominal conditions
(a) Head level states of various nodes under attack conditions, with the dashed line representing the same time point as described in Figure 4.3a. The difference is that an attacker is now manipulating speed set-points for the booster pumps on this system, dramatically affecting water availability.

(b) Head level of the entire system after 48 hours under attack conditions, showing that all nodes are similarly affected by the attack, resulting in low head-levels.

Figure 4.4: Simulation of the water system under attack conditions
In summary, this work has demonstrated two case studies on analyzing vulnerabilities in critical infrastructure systems. The stated objectives of this thesis were to:

1. Illustrate a spying attack on an unmanned aerial vehicle swarm, along with how the problem could be mitigated by using secure multiparty computation,
2. Present an attack to disrupt the service of a hypothetical municipal water system, through the control of speed settings on booster pumps,
3. Demonstrate the use of system models to highlight inherent system vulnerabilities.

In the first case study, we performed simulations of a UAV swarm conducting a search-and-rescue mission under two scenarios, one with the swarm being vulnerable to a spying attack and the other with the system being protected against such an attack. The results of the former scenario showed how an adversary could estimate the locations of all agents in this swarm through compromising a single member of the swarm. In the latter, the system observability was reduced through the use of secure multiparty computation which eliminated the ability of the adversary to estimate the locations of the uncompromised drones.

The second case study presented a denial-of-service attack on a hypothetical municipal water system. The novelty of this portion is in the use of models of the booster pumps which maintain head levels in the network. Previous work utilizes pump curves to simulate the effects of these pumps [46], while the results in this thesis use pump models to show the effects of attacking the speed settings on booster pumps. These models were validated against
the input-output behavior specified by the pump curves under normal operating conditions. An initial simulation of the integrated water system showed the nominal behavior of the system in adequately distributing water throughout a municipality. The second simulation highlights the effects of hijacking the booster pumps to be able to drastically reduce head levels and water availability in the system.

Lastly, we demonstrate the use of system models to highlight inherent system vulnerabilities within both case studies. While the provided example systems have vulnerabilities that are relatively straightforward to understand and analyze, this offers a foundation for finding similar vulnerabilities that can be obscured by more complex networks in these domains.

There are a number of avenues for extensions of this work and future research that can be conducted. In regards to UAV drone swarms, this work only looked at observability attacks in the context of a single formation, from an insider threat, and with only a single exposed vehicle within the swarm. Other attack forms, like controllability or denial-of-service attacks within the contexts of different objectives, vulnerable vehicles, formations, and attacking perspectives can all be modified to identify further system vulnerabilities. Likewise, there are similar areas that can be expanded when considering municipal water systems. Linear approximations of the nonlinear dynamics of the water network models, as defined in [58], can be used to define the Dynamical Structure Function (DSF) [21]. The DSF can then be used to identify specific aspects of the network that can lead to these attacks, under the assumption that the specified points can be observed, controlled, or otherwise manipulated. Aside from the linear approximations, techniques similar to those described by [27] can be used to define the DSF from either nonlinear approximations or the nonlinear models themselves.

Aside from the existing water infrastructure, this analysis could also be extended other aspects on infrastructure on which the water system depends or which depend on the water networks. Additionally, scenarios such as fire prevention and suppression, or critical cooling systems, such as those found in nuclear power generation facilities, could be further areas of research.
References


