Spring 2015

Securing communication within the harms model for use with firefighting robots

Maxwell D. DeWees
Purdue University

Graduate School Form 30
Updated 1/15/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Maxwell DeWees

Entitled
SECURING COMMUNICATION WITHIN THE HARMS MODEL FOR USE WITH FIREFIGHTING ROBOTS

For the degree of Master of Science

Is approved by the final examining committee:

Eric T. Matson
Chair

Anthony H. Smith

J. Eric Dietz

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Eric T. Matson

Approved by: Jeffrey Whitten 4/20/2015
Head of the Departmental Graduate Program Date

SECURING COMMUNICATION WITHIN THE HARMS MODEL FOR USE WITH FIREFIGHTING ROBOTS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Maxwell D. DeWees

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2015

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

Thank you to my parents, George and Tamara DeWees, for supporting me whenever needed, for motivating me, and for creating an environment that always inspires greatness.

I also want to thank my advisor Eric Matson for providing support and allowing me to pursue this thesis, as well as Professors Eric Dietz and Tony Smith for forming my committee and offering guidance.

Finally, a special thanks to my peers and loved ones Ashley Ancil, David Hersh, Jake Kambic, Lucas Starrett, and John Wallrabenstein for all of your helpful contributions. Without all of you, this would not have been possible.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
CA	Certificate Authority
CBC	Cipher Block Chaining
CDH	Computational Diffie-Hellman
CPU	Central Processing Unit
DDH	Decision Diffie-Hellman
DHKA	Diffie-Hellman Key Agreement
DHS	Department of Homeland Security
FBI	Federal Bureau of Investigation
FPRA	Fire Protection Research Foundation
HARMS	Humans Agents Robots Machines and Sensors
HKDF	HMAC-based Key Derivation Function
HMAC	Keyed-Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
HVAC	Heating, ventilating, and air conditioning
ICS-CERT	Industrial Control Systems Cyber Emergency Response Team
IEEE	Institute of Electrical and Electronics Engineers
IFF	Identification, friend or foe
IKE	Internet Key Exchange
IP	Internet Protocol
IPSec	Internet Protocol Security
MAS	Multi-Agent System

MD5	Message-Digest Algorithm 5
MitM	Man-in-the-middle
NFFP	Navy's Fire Fighting Project
NFPA	National Fire Protection Association
PDD	Presidential Decision Directive
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
PSK	Pre-shared key
RAM	Random Access Memory
SAFFAR	Security and Fire Fighting Advanced Robot
SAFFiR	Shipboard Autonomous Firefighting Robot
SCADA	Supervisory control and data acquisition
SHA	Secure Hash Algorithm
TCP	Transmission Control Protocol
VM	Virtual Machine

GLOSSARY

adversary	any unauthorized entity that exists on the network, capable of listening, capturing, or modifying network traffic
authentication	the property of validating that an entity is in fact who they say they are
broadcast message	in the HARMS system, a message that is sent to every peer in a system actor's peer list
confidentiality	the property of preventing unauthorized entities from viewing the contents of a message in a meaningful way
ciphertext	text that has been transformed in such a way that makes it difficult to comprehend the original message; see confidentiality
critical infrastructure	"the assets, systems, and networks, whether physical or virtual, so vital to the United States that their incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof" ("What is critical infrastructure?", 2013, p. 1)
emergent behavior	a property where simple systems join together in a complex environment to exhibit more complex behaviors (Russell & Norvig, 2009)

HARMS model a layered model where humans, agents, robots, machines, and sensors connect, communicate, and interact in a decentralized ad-hoc environment for task completion (Lewis et al., 2013)

HARMS system the software implementation of the HARMS model

indistinguishability in the HARMS model, a system actor is not concerned with the physical makeup of an actor it is communicating with, only in its ability to solve a goal or execute a task (Matson & Min, 2011)

integrity the property of being trustworthy; for data, this property is held if the data has not been modified

man-in-the-middle attack a network attack where the adversary exists in the middle of two communicating parties and can listen to, capture, or change the data in some way

multicast message in the HARMS system, a message that is sent to a specified amount of peers in a system actor's peer list

plaintext the original, unmodified contents of a message

replay attack a network attack where the adversary can capture a message and send it at an arbitrary point in time to the original recipient

system actor in the HARMS model, any human, agent, robot, machine, or sensor member of the network

unicast message	in the HARMS system, a message that is sent to one peer in a system actor's peer list

ABSTRACT

DeWees, Maxwell D. M.S., Purdue University, May 2015. Securing Communication Within the HARMS Model for Use with Firefighting Robots. Major Professor: Eric T. Matson.

Humans and robots must work together in increasingly complex networks to achieve a common goal. In this research, firefighting robots are a part of a larger, decentralized system of humans, agents, robots, machines, and sensors (HARMS). Although communication in a HARMS model has been utilized in previous research, this new study looks at the security considerations of the communications layer of the HARMS model. A network attack known as a man-in-the-middle attack is successfully demonstrated in this paper. Then, a secure communications protocol is proposed to help provide confidentiality and authentication of HARMS actors. This research is applied to any system that utilizes a HARMS network, including firefighting robots, to help ensure malicious entities cannot exploit communications by system actors. Instead, system actors that confirm their identity can communicate securely in a decentralized way for indistinguishable task completion. The results of this experiment are successful, indicating that secure communication can prevent man-in-the-middle attacks with minor differences in operation.

CHAPTER 1. INTRODUCTION

Multiple humans, software agents, robots, machines, and sensors (HARMS) can join together to create a network of users or system actors to work together toward a common goal. In this HARMS model, emergent behavior is observed as indistinguishable, meaning any actor who is capable of performing a certain task is chosen to do so, regardless of other factors such as architecture or cognitive design. In this scenario, communication between HARMS actors is essential to other modes of operation and can be achieved in a number of ways. The collection and analysis of data in a robotic network is typically shared with other actors, which could be other robots, agents, or humans in the system. Before this project, there was no mechanism for providing authentication to the network for authorized users or protecting the data being transmitted. This opened up vulnerabilities in the network for adversaries to communicate with HARMS actors in an unauthenticated manner. Therefore, the major goal of this research was to provide authorized, secure communication in a multiagent network while maintaining indistinguishability.

This research applied this goal toward firefighting robots participating in a HARMS-model network. Firefighting robots have already been successfully used to help aid human first responders. When a command is given to a firefighting robot, either from one robot to another or from a human actor, this command must be authenticated and sent to the correct robot at all times. Communications should not be intercepted, altered, or replayed by adversaries in an emergency response situation. Unique to this situation, however, was securing communication while maintaining emergent behavior and allowing for an automated decision-making process among one or more actors. This provides indistinguishable task completion, a major goal of actors in any HARMS model network, and a behavior that is especially important in critical infrastructure networks such as multiagent firefighters.

1.1 Scope

In this project, a multiagent robotic network refers to a HARMS-model network (Lewis, Matson, Wei, & Min, 2013), consisting of any number of humans, agents, robots, machines, and sensors. Each component of a HARMS-model network is referred to as a system actor. For this work, the network was composed of one or more actors of similar or different types with the ability to communicate with each other. In other words, the actors were heterogeneous in that they do not have to be of similar design, architecture, shape, or ability. At a minimum, each actor needed to have some mechanism for communication with the other actors in the network. Furthermore, the HARMS model provides a goal of indistinguishability, where any actor who is capable of performing a certain task is chosen to do so, regardless of the other factors previously mentioned.

Actors or users are said to be authorized if they have permission to be a member of the network. This permission can be given explicitly by the owner of the network or through authentication mechanisms, which will be discussed later. In contrast, unauthorized or malicious actors do not have permission to be a part of the network. Any communication by unauthorized actors is unwanted and is seen as malicious behavior or an active attack.

Communication between actors can be performed in several different ways, including standard Internet protocols used between machines, robots, and agents, as well as natural language (e.g., text or speech) used by humans. Within the scope of this project, communication between agents, robots, and machines were attempted to be secured. Specifically, exploitations known as man-in-the-middle attacks were considered. As discussed in Chapter 2, a variant of the man-in-the-middle attack called the MiG-in-the-middle (Anderson, 2008) was examined. In this attack, the concept of identification, friend or foe (IFF) is crucial, where authorized actors are correctly distinguished from unauthorized ones. A subset of the man-in-the-middle attack, known as a replay attack, was also considered and is discussed in this project. Other attacks were considered outside the scope of this project and left for future research.

The development of security mechanisms for communication in a HARMS-model network was applied to firefighting robots. This application provided a real-world scenario where a HARMS-model network could be used. Specifically, multiple human firefighters and multiple firefighting robots create a network whose goal is to extinguish a fire. These firefighters are considered the authenticated users, because the humans need to communicate to the robots to control them via wireless remote control. Firefighting robots also have the ability to communicate with one another.

1.2 Significance

Although the HARMS model was developed in previous work, no mechanism for securing the communication of system actors in a HARMS-model network existed previously. Secure communication from certain attacks was novel in a HARMS-model network, because it provided decentralized authentication and confidentiality while maintaining indistinguishability. This means that authorized actors can communicate securely with other authorized actors, but the overall goal is still performed without the direct request of a user of the network to a specific actor. An authorized system actor does not need to perform special (or inconvenient) steps to communicate securely, but adversaries are unable to understand the communications or participate without authorization.

Additionally, applying security of communications to a firefighting robot is important because adversaries could cause significant damage, including loss of life, if they are able to successfully tamper with or disrupt communications to the robots during a firefight. The attack vector looked at specifically in this research, generalized as a “man-in-the-middle” attack, is commonly seen in many applications, including banking, e-commerce, and military environments (Anderson, 2008). Therefore, it is inevitable that adversaries will attempt this well-known attack to leverage firefighting robots should they be relied on as the primary mechanism for firefighting. This research could also be applied to other disaster recovery or emergency response scenarios where using robots or HARMS-model networks are also appropriate.

1.3 Research Question

In a multiagent robotic network, can communication between authorized users be secured from unauthorized or malicious users?

1.4 Assumptions

The assumptions for this study included:

- Authenticated system actors were known at all times.
- Other wireless communication technology were not interfering or transmitting during the experiment unless part of the designed attack.

1.5 Limitations

The limitations for this study included:

- Only man-in-the-middle (or middleperson) attacks and replay attacks were considered.
- The solution is generalizable to all HARMS systems using for unicast, multicast, and broadcast messages.

- Implementation of secure communications might affect the speed of communication or the work required to authenticate users.

1.6 Delimitations

The delimitations for this study included:

- Other types of network attacks (including but not limited to brute-force/exhaustive key search to break encryption, denial of service, or other side-channel attacks) were not considered.
- Physical attacks such as tampering with or removing the robot or social engineering were not considered.
- For safety reasons, experimentation did not occur during an actual fire.
- Moral, ethical, or philosophical questions regarding the use of firefighting robots were not considered.

1.7 Summary

As robotics begin to integrate into service tasks such as firefighting, it is becoming increasingly more important for minimum security assurance levels to be present. As with any technology used in critical infrastructure, health, or public safety, potential cyberphysical vulnerabilities need to be identified and mitigated before they are used to cause damage or in other malicious ways. This chapter provided the significance of this research project, which allows the scope to be drawn around firefighting robots in a HARMS network. Although many attacks are potentially feasible, the focus for this project was to secure communication against man-in-the-middle and replay attacks. Assumptions, limitations, and delimitations were provided to help describe boundaries and other issues that were expected to be encountered. The next chapter provides a review of the background literature relevant to this project.

CHAPTER 2. REVIEW OF RELEVANT LITERATURE

This chapter provides a review of the background literature relevant to the advancement of robotics with firefighting capabilities, both as individual robots and working together in multiagent robotic networks. This research looks at the domain of firefighting robots as it relates to a conceptual model for humans, agents, and robots to work together. This model provides a compelling platform for multiple firefighting robots of potentially different designs to cooperate to accomplish a common goal of extinguishing fires and eliminating fire threats to both victims as well as human first responders.

However, communication between humans and robots, as well as communication between robots themselves, needs to be secured so that unauthorized users or adversaries cannot inflict damage to the robots directly or use them maliciously. Firefighting in general is a subset of critical infrastructure, or services and capabilities that are core to a country, and securing critical infrastructure against cyberphysical attacks is a significant but complicated issue. Therefore, this chapter will also provide a discussion of security in robotics and critical infrastructure, primarily from a viewpoint within the United States.

This chapter will provide a basis for the questions identified in the first chapter and explore how this project can attempt to solve this problem. This chapter will be split into three major sections: background on firefighting robotics, information on the network attacks considered for this research, and an overall view of security and policy in the United States. This chapter will also serve as a starting point for highlighting previous, related research and provide a history and background information relevant to the project.

2.1 Introduction to the HARMS model

Research and development in robotics have increased the capabilities of robots and software agents rapidly. Humans are beginning to rely more and more on the use of robots for task completion of many kinds, including in the workplace, as municipal and private services, and even at home. The HARMS model (Matson & Min, 2011) was developed to help create a system to bring robots and machines together with humans so that they can cooperate or perform as a single entity or collective organization. The HARMS model provides mobility, self-organization, scalability, adaptability, and indistinguishability to a decentralized network of (H)umans, (A)gents, (R)obots, (M)achines, and (S)ensors. Any one of these members of a HARMS-model network is referred to as a system actor. Furthermore, a benefit of using the HARMS model is that it provides flexibility among configuration of system actors. A network could be composed of many of one type of system actor, or several different system actors working together. These combinations will be of use to firefighting robots and will be discussed in detail later in this chapter.

The HARMS model is layered such that each layer includes and transcends the previous one (Lewis, Matson, Wei, & Min, 2013). These layers start with Network, the most fundamental layer and build up to Collective Intelligence, where the model strives to provide emergent behavior via a collection of one or more agents, robots, and humans (see Figure 2.1 for more details). Through this model, the goal of indistinguishability is enabled, where any actor who is capable of performing a task is chosen to do so without preference on which system actor actually performs the task and regardless of build, architecture, or behavior. This model is decentralized in that each system actor communicates directly with one or more other actors directly, rather than communicating through a fixed point. A HARMS-model network provides the ability to send messages to multiple system actors at once (known as multicast) or to the entire system (known as broadcast).

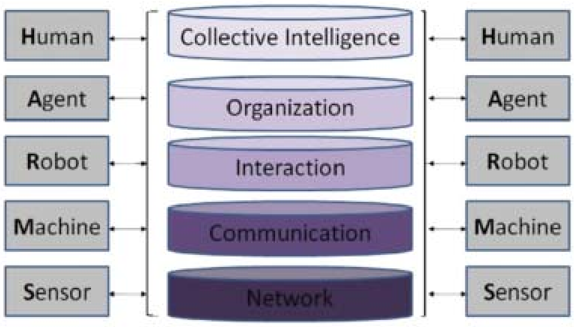


Figure 2.1.: The HARMS layered model

Each layer of the model will be explained briefly. First, the Network layer "represents the basic communication between system actors. Each system actor must have basic capabilities to connect to other actors" (Lewis et al., 2013, p. 1187). The second layer, Communication, "enables the basic common exchange capability between any systems actors. Communication is defined by elements such as meaning, syntax, protocols, and semantics." It is important to note that this research exists between the first two layers of the HARMS model. Then, the Interaction layer "represents a set of commonly developed algorithms and techniques which provide a layer for group rational decision making". The Organization layer "uses multiagent systems organization models" to provide roles to accomplish one or more goals. Finally, the Collective Intelligence layer "will not only allow emergent behaviors, but also the connection of multiple organizations into higher-level collectives such as societies or organizations, and potentially a definition of consciousness" (Lewis et al., 2013, p. 1187).

The HARMS model provides a real-world basis for which multiagent robotic networks can be assembled for task completion. Previous research has been successful in creating mobile wireless mesh networks in disaster areas to help provide relief (Nguyen et al., 2012). Rescue robots can utilize the HARMS model very effectively, because there might be robots of different sizes and designs needed to provide disaster relief or rescue operations simultaneously. In other words, one type of robot alone may not be sufficient in providing assistance on an adequate level. Firefighting robots provide a good example of this and will become the focus for this thesis. The next section will provide a brief overview of firefighting robots, both used individually and as a part of a larger network. Firefighting robotic networks such as the one developed by Min et al. (2014) will be studied in more detail in the following sections.

2.2 Introduction to Firefighting Robots

The use of robotics in firefighting applications is a relatively recent advancement. This section will provide details on the history of firefighting robots, the research that was done to develop them, and how they are used today. This includes commercial, military, and research applications. Furthermore, the concept of using multiple firefighting robots together in a single fire event is discussed. These multiagent firefighting robotic networks are important to keep in mind, as they provide significant security implications.

2.2.1 History & Research

Although the concept of a firefighting robot was first mentioned in the early 1960s (Thring, 1963), the first functional robot to combat fire appeared twenty years later (Kobayashi & Nakamura, 1983). This project, lead by a Japan Industrial Robot Association (JIRA) committee, defined several functions for the robot: inspection, refuge guidance, and rescue work. Based on these design decisions, the committee designed a small ground vehicle with two parallel continuous tracks, or tank treads in differential drive configuration. One such design was controlled via wireless radio waves from a human controller (see Figure 2.2). Another potential design included a microphone, speaker, and wide-angle lens. This is significant because the tank-like ground vehicle robot design (typically with some sort of fire-hose attachment or extinguisher mechanism) has become the widely-adopted standard design for firefighting robots.

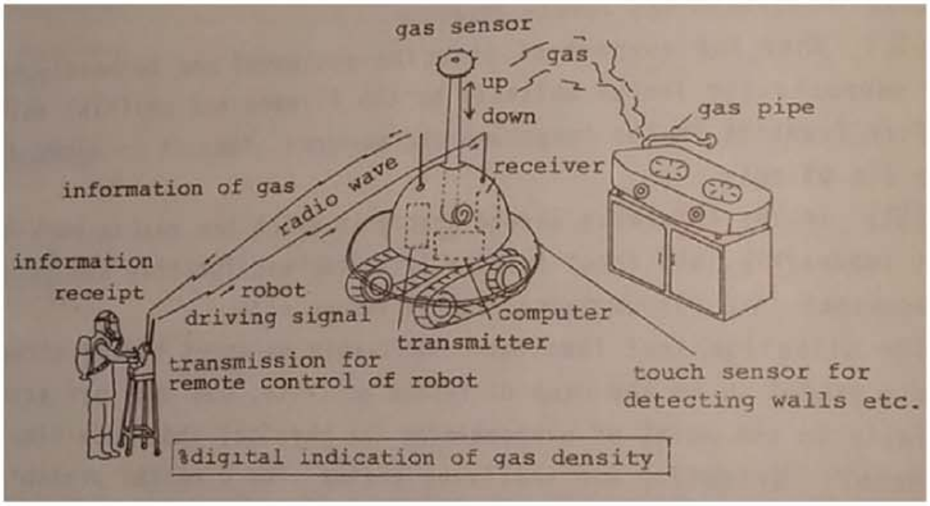


Figure 2.2.: A preliminary look at firefighting robots from 1983 (Kobayashi & Nakamura, 1983)

A few years later, the United States Navy became interested in a firefighting project involving the use of remotely-controlled vehicles or robots (D. L. Smith, Petroka, Yobs, Lewis, & McCarthy, 1985). This became known as the Navy's Fire Fighting Project (NFFP). Smith et al. proposed a system to not only replace a human firefighter but to outperform the firefighter by using engineering design principles. Two fire extinguisher mechanisms were proposed, for example, using both fire hoses and fire extinguishers.

In 1991, researchers introduced the United Kingdom's Security and Fire Fighting Advanced Robot (SAFFAR), a project created to help address fires caused as a result of arson (Bradshaw, 1991). In this research, a modular design was proposed so that a product line or "family" of robots could be built with varying cost and performance parameters. A product line of robots would in theory be compatible with each other, hinting toward building a network of working robots. The article concluded that the next stage for firefighting robots would be autonomous and mobile control. Indeed, autonomous navigation was quickly introduced as a design goal in subsequent research. A competition at New Mexico Institute of Mining and Technology was started in 1999 to create an autonomous robot that could navigate a maze and extinguish a candle (Schumacher, McVay, & Landes, 1999). Although this is arguably not a “firefighting robot,” many research and academic projects like it emerged. One such example is the 2003 IEEE SoutheastCon Hardware Competition, where students had to build autonomous robots to find and extinguish simulated fires (Dubel, Gongora, Bechtold, & Diaz, 2003).

Firefighting robots are not always vehicular, however. Researchers in Norway designed a snake-like (or hose-like) robot called Anna Konda, which has the water pressure to break walls (Bless, 2006). Another recent example is the humanoid robot developed by the Naval Research Laboratory, the Shipboard Autonomous Firefighting Robot, or SAFFiR. These researchers, in cooperation with Virginia Tech and the University of Pennsylvania created a humanoid firefighting robot to “enable more robust performance in difficult environments” as it attempts to mimic the ways humans walk and operate as firefighters (Lahr, Orekhov, Lee, & Hong, 2013, p. 1).

2.2.2 Commercial Firefighting Robots

Soon after research began on firefighting robots, fire departments began using them to help combat real-world fires. A market was created for commercial firefighting robots, and several robots were quickly introduced for individuals and municipalities to purchase worldwide. The Tokyo Fire Department, where some of the earliest firefighting robotics research began, employs 12 different firefighting robots as of 2011, including the Robocue, which is a large tank-like, vehicular robot used in rescue operations to save people and move large objects (Heller, 2011). A robotics company based in Croatia, DOK-ING, offers a large remote-controlled firefighting robot, the MVF-5 (“MVF-5”, n.d.). The MVF-5 allows a human user to operate it remotely from up to 1,500 meters away, while providing six video feeds that the operator can control. This particular robot, pictured in 2.3, also allows supports different water hookup sources, including from fire trucks and fire hydrants.

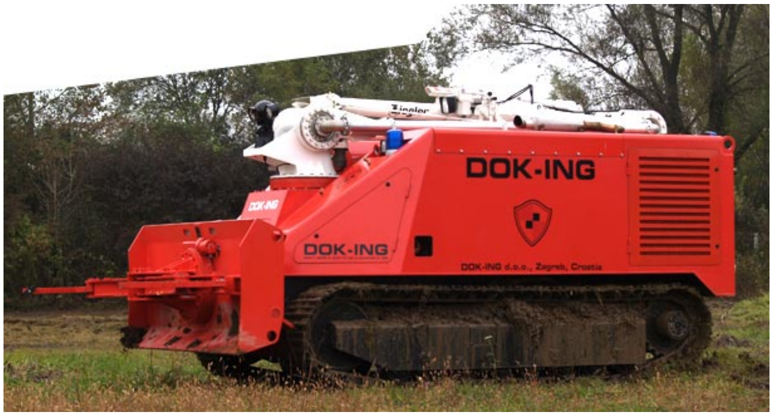


Figure 2.3.: The DOK-ING MVF-5 firefighting robot (“MVF-5”, n.d.)

Another commercially available robot, the Howe and Howe Technologies' Thermite (Plackett, 2012), has the ability to extinguish fire using a 600 gallon-per-minute hose and costing around $97,000. This robot, which started as a U.S. Department of Homeland Security project, became commercially available in 2012. Recently, fire departments in the United States have adapted use of firefighting robots in various capacities. During a wildfire around Yosemite National Park in 2013, the National Guard used an unmanned aircraft to help provide aerial views of the park (Skoloff & Cone, 2013). Although not specifically a firefighting robot, the use of unmanned vehicles to aid in firefighting has quickly grown in popularity. Seen in early 2015, firefighters in Arlington, Texas, used unmanned hoses to help cool the source of a fire and used a robot to observe flames and heat levels (Davis, 2015).

Examples like these, along with cases of firefighting robots used in Alaska and Oregon (B. Smith, 2014) seem to indicate a general trend toward increasing the use of robots, drones, or other unmanned vehicles to provide aid or completely replace human firefighters. Indeed, it is the goal of the Fire Protection Research Foundation (FPRA), a research arm of the National Fire Protection Association (NFPA) to have "all firefighting apparatus and equipment used by emergency responders. . . not in physical contact with the individual when operational"(Grant, 2014, p. 52).

2.2.3 Multiagent Firefighting Robots

Once robots successfully demonstrated that they could perform certain basic tasks, research increased to more advanced tasks and began exploring the possibility of using more than one robot at once to achieve a common goal (Weiss, 1999). Multiagent robots or Multi-Agent Systems (MASs) provided a way to distribute the workload to not only perform tasks more quickly than before, but also taking advantages of using design differences in certain individual robots to decide which agent was best suited for a particular sub-task, allowing MASs to also complete more complicated tasks. When using multiagent robotics, organizing an authority hierarchy is important (Esmaeili, Mozayani, & Motlagh, 2014), because it creates social organizations similar to humans. Of course, multiagent robotics was soon applied to the domain of firefighting.

By the turn of the century, academic and government research identified problems with firefighting robots. Hisanori Amano, a researcher at the National Research Institute of Fire and Disaster in Japan noted that although fire departments had already begun to utilize firefighting robots, the current robots were not designed with fire department needs in mind (Amano, 2002). Many of them were too large or weighed too much, causing mobility issues and limiting them in certain areas. Furthermore, these robots were very expensive due in part because private companies were not interested in developing robots for a small, niche market. Amano concluded that the next generation of firefighting robots would be not one solution but a group of robots of various sizes and functions utilized by first responders (Amano, 2002).

A couple of years later, research was presented on a network of firefighting robots. The authors envisioned "a physical network that can sense, move, compute, and reason, letting network users (firefighters and first responders)" to search for information about the environment (Kumar, Rus, & Singh, 2004, p. 24). Their initial experiment used a small network of sensors with radio tags that the robots can communicate with to localize and build a map of the room. The robots communicated with each other using the IEEE 802.11b wireless networking specification, however it is not mentioned whether or not the communication was encrypted or secured in any way.

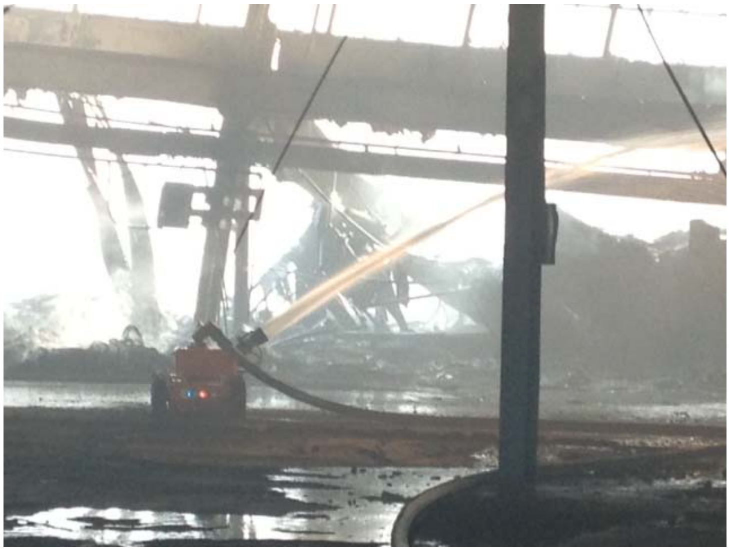


Figure 2.4.: The Dongil field robot FIRO-M combating fire in Hoopeston, Illinois (Min et al., 2014)

Related to this project, previous research with multiagent firefighting robots has been successful in combating real-world fire. This research allows HARMS model networks to be created for disaster relief situations. It was put to the test in July 2013 in Hoopeston, Illinois, with a large-scale fire of a tire recycling plant, some 400,000 sq. ft. large. Researchers sent in a Dongil Field Robot FIRO-M (pictured in 2.4), which was able to reach places within the building that were not accessible to human firefighters (Min et al., 2014). In order to create a multiagent network, a leader role is assigned to one of the robots, while the others become followers. The leader “computes navigation trajectories to create the network” which is then communicated to the follower robots. The communication here is done via the IEEE 802.11 wireless specification, but security of the communication is not considered. This project concluded that the “results show promise for developing quickly configured networks” for use with buildings such as the one in Hoopeston, Illinois (Min et al., 2014, pp. 6).

A group of European researchers looked at a slightly different subset of firefighting robots working together in a network. This project involved “swarm robotics,” which differs from a multiagent robotic network because all robotic members are homogeneous, or of the same type and build and the structure of the network is decentralized. These robots are small and somewhat underpowered when looking at a single robot’s abilities, but work together to provide a large amount of telemetry and useful statistical data to first responders during an event (Naghsh, Gancet, Tanoto, & Roast, 2008). The authors state that one of the problems they encountered with the project was a communication overhead, where members of the swarm had to confirm the position of various other robots throughout the task. This highlights an important potential vulnerability for robotic networks. Because having reliable information on where members are at any given time is critical in an emergency response scenario, adversaries who can deceive these robots into accepting incorrect or falsified information would severely impact these robots’ abilities.

Indeed, there are several security concerns when dealing with swarm robotics or multiagent networks. In particular, wireless communication using radio waves are susceptible to interception or tampering (Higgins, Tomlinson, & Martin, 2009). Convincing robots or humans of individual identity or group identity can also be potentially difficult, and results in an identify-friend-or-foe (IFF) situation. According to these researchers, robots used in disaster relief have a primary security requirement of availability:

> If robots are unavailable due to malfunction, accident or because they have been hijacked either physically or electronically by an external agency, then they will be unable to perform their critical task. Confidentiality could be necessary to safeguard information about entities that the robots come across, such information could be highly sensitive or of other interest to malicious parties (Higgins et al., 2009, pp. 310-311).

Although swarm robotics differ slightly from multiagent robotic networks, it is still important to consider these security challenges. The next section of this literature review will begin to detail one such attack vector that this project hopes to address.

2.3 The MiG-in-the-Middle Attack

The man-in-the-middle attack, previously mentioned in the last chapter, is a common attack vector whenever two or more actors or systems communicate with one another. A man-in-the-middle attack is performed when an adversary can intercept or capture communication from one of the parties without the awareness of any party involved ("Man-in-the-middle attack", 2014). The adversary may not do anything other than capture and record the communication which otherwise is assumed to be private, or the adversary may attempt to actively alter the communication as it travels from one party to the other. This attack can be performed in many different situations, including wireless communication between systems and Web traffic. It is important to note that encryption alone cannot stop a man-in-the-middle attack, which will be discussed in more detail shortly.

Information security researcher Ross Anderson talks not about robotic networks, but military air defence forces. In order to develop a system to identify-friend-or-foe (IFF), he states: “the typical air defense system sends random challenges with its radar signals, and friendly aircraft have equipment and keys that enable them to identify themselves with correct responses” (Anderson, 2008, p. 73). However, one way to circumvent this solution, which was demonstrated in real-world combat situations, was for adversaries to capture the correct responses and replay them to the air defense system as their own. Specifically, if the adversary can place himself or herself in between the two authorized points of communication and relay messages from one to the other, he or she can perform a man-in-the-middle attack.

The attack as Anderson describes it is as follows: South African forces were fighting a war in the 1980s in northern Namibia and southern Angola, with Cuban forces helping their Angolan allies. Cuban forces, flying MiG aircraft, were nearby a South African air base. When the South African bombers left there to attack an Angolan target, the Cuban MiGs flew through the South African air defenses where they could receive South African IFF challenge messages that were encrypted. The MiGs sent them to the Angolan defense, who was presently engaged in combat with the South African bombers, and the Angolans broadcast these IFF messages out. The South African bombers sent their automated responses back, since their IFF equipment was left on and were in the appropriate vicinity. The Angolans were able to relay the responses back to the Cuban MiGs; the Cuban MiGs could now answer the IFF challenge correctly and were therefore left untouched, where they were able to carry out a successful bombing raid (Anderson, 2008).

This story, whether true in South Africa or elsewhere, illustrates an important point. Implementing sound cryptographic techniques to secure communication does not guarantee that the system is actually secure. In the MiG-in-the-middle example, the Cuban MiGs had no idea what the IFF challenge response was. They just understood that if they could capture a correct response, they could replay it and the system would accept it. The next section will discuss the extent to which these network attacks occur and how costly they can be.

2.4 Prevalence of Network Attacks

Ross Anderson’s story of the middleperson attack in South Africa is not the only successful example of the attack during combat. In World War II, German bombers would shut off their transmitters during air raids. The British then turned on their high power transmitter, called Aspidistra, and began transmitting on the same frequency as the bombers would have used if they were on. The British would start by simply retransmitting the German network broadcast occurring from another source, but then would quickly change to convincing but false pro-Allied propaganda. Due to how authentic the transmission sounded, many German personnel believed them, causing confusion and even convincing “people to evacuate to seven bomb-free zones in central and southern Germany” (Schneier, 2008, p. 1).

Perhaps the most well-known and widely-discussed example of malware used in cyberwar, the Stuxnet worm, also uses a middleperson attack in one component of its complicated process. According to Larry Constantine, an author in the fields of computer science and cybersecurity, Stuxnet was able to cause damage to so many of Iran’s nuclear centrifuges due to a man-in-the-middle attack it performed against the industrial control system (Cherry, 2011). This was corroborated for an older version of Stuxnet in a Symantec report from 2013 (McDonald, Murchu, Doherty, & Chien, 2013).

The Electronic Frontier Foundation reported a middleperson attack against Google in Iran in 2011. Here, a certificate authority issued a certificate to an adversary for an Iranian Google page. This meant that although users tried to access Google with HTTPS, an encrypted version of the web page, it was not actually Google but rather a malicious third-party. This third-party was able to convince Iranian users to log in to their email accounts and perform potentially sensitive searches while they were able to intercept all traffic to and from Google (Schoen & Galperin, 2011).

Attacks against specific nation-states or for specific political reasons are on the rise. These attacks might leverage existing services such as Google or mobile phone networks, but attacks against critical infrastructure are also gaining popularity. In 2013, researchers at the Technical Unit for Energy and Environmental Modeling in Rome, Italy modeled certain cyber attacks on components of critical infrastructure they identified. In fact, man-in-the-middle attacks are one of the most popular vectors for adversaries and the authors discuss the potential consequences from such an attack (Ciancamerla, Minichino, & Palmieri, 2013). Specifically, as much as 30% of the system was found to be affected by a middleperson attack at the end of their experiment, modeled and exploited with relative ease and success. This highlights a large problem with critical infrastructure security, which is discussed in detail in the next section. Since fire departments and first responders of situations involving fire are considered a part of critical infrastructure, it is significant to consider what may happen in the near future.

2.5 Security in Critical Infrastructure and First Response Technology

In 1998, the President of the United States released a Presidential Decision Directive calling for the first time, protection of critical infrastructure (*The Clinton Administration's Policy on Critical Infrastructure Protection: Presidential Directive 63*, 1998). This document states that critical infrastructure, including telecommunications, transportation, and emergency systems such as police and fire services have growing potential vulnerabilities. Furthermore, the White House stated:

> Because of our military strength, future enemies, whether nations, groups or individuals, may seek to harm us in non-traditional ways including attacks within the United States. Because our economy is increasingly reliant upon interdependent and cyber-supported infrastructures, non-traditional attacks on our infrastructure and information systems may be capable of significantly harming both our

> military power and our economy (*The Clinton Administration's Policy on Critical Infrastructure Protection: Presidential Directive 63*, 1998, p. 1).

After the attacks on the World Trade Center buildings in New York on September 11, 2001, the United States created the Department of Homeland Security "to detect, prepare for, prevent, protect against, respond to, and recover from terrorist attacks within the United States" (*National Strategy for Homeland Security*, 2002, p. 1). Within the Department of Homeland Security, the Office of Infrastructure Protection was created. According to the Department of Homeland Security's website: "The office conducts and facilitates vulnerability and consequence assessments to help critical infrastructure owners and operators and State, local, tribal, and territorial partners understand and address risks to critical infrastructure" ("Office of Infrastructure Protection", 2014, p. 1). Furthermore, the President updated PDD-63 by releasing the Homeland Security Presidential Directive 7, where the Secretary of Homeland Security is "responsible for coordinating the overall national effort to enhance the protection of the critical infrastructure" (*Homeland Security Presidential Directive 7*, 2003, p. 2).

Indeed, the United States federal government seems to be right to worry. As new cyberwar abilities continue to develop, the overall trend in attacks to critical infrastructure is rising. In 2012, the number of attacks reported to the Department of Homeland Security's Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) grew by 52% (Goldman, 2013). Recent research stated that nearly 70% of critical infrastructure companies (those that provide "power, water, and other critical functions") reported one or more security breaches in 2013 alone (Unisys, 2014).

A specific attack reported by ICS-CERT in 2012 was against the computer networks of natural gas pipeline companies in the United States. It appeared to start as early as December 2011 and was primarily conducted through targeted phishing emails sent to personnel within these companies (Brenner, 2012). More recently, a cyberattack on a steel mill in Germany was able to cause physical damage to the plant via remotely controlling and disrupting critical systems (Zetter, 2015). These examples illustrate the variety of attack vectors possible against critical infrastructure and how necessary it is to begin securing our critical infrastructure technology today.

2.6 Summary

This chapter provided a look at pre-existing research done to develop the HARMS model, a way to effectively build a network of multiagent firefighting robots. As we went through the history of robots designed for firefighting, we learned that they can largely benefit from working together with many robots in a network, but this concept introduces security vulnerabilities that have been previously seen and exploited successfully. Therefore, it is important to secure this network now, rather than waiting until it is an afterthought. Firefighters and firefighting robots are considered critical infrastructure, and it is crucial to secure our nation's infrastructure against attack, because without it, there are significant consequences to our economy, safety and well-being.

Unfortunately, attacks against critical infrastructure are on the rise and are projected to continue to do so. Therefore, it will only become more difficult to prevent adversaries from successfully breaking into technology that we depend on as our nation's backbone. However, it is still early on, and as echoed by Higgins et al. (2009), little work has been done to previously secure networks of multiagent robotics. The next chapter provides the framework and methodology to be used in this research project.

CHAPTER 3. FRAMEWORK & METHODOLOGY

This chapter provides the framework and methodology used in the research study. Details on the implemented solution are provided, including justification on why this solution was chosen. The hypotheses are presented, and further information on testing is provided. This sets up the ability to test this solution and provide results and analysis in this next chapter. This chapter will also include the measure for success, variables to consider, and methods of data collection. A framework for this research is discussed for repeatable future experiments.

3.1 Research Approach and Hypotheses

This project was a quantitative study on the feasibility of securing communication between system actors of a HARMS-model network. As discussed previously, a system actor is a human, software agent, robot, machine, or sensor participating in a HARMS network. This project main goal was to secure communication of any HARMS-model actors, including a primary focus on an implementation for firefighting robots, developed in previous research. The apparatus for this project included multiple machines, such as computers and robots that are compatible or comparable to the previously developed firefighting robots (used as a proof-of-concept). The technology used for communication, including any remote controls or wireless technology, communication protocol, and security mechanisms was also a part of the apparatus. These communication details are generalizable to any robotic network using the HARMS model.

Furthermore, technology to capture wireless traffic was utilized during the experiment phase of this research. This was used to simulate an adversary's attempt to capture traffic from a member of the network, either to learn information about what was sent or to perform an active man-in-the-middle or replay attack. It is important to distinguish between plaintext and ciphertext messages. Plaintext messages can be in natural language text for human readability or in text/binary form for machine computation. On the other hand, ciphertext is the encrypted version of the plaintext. It is not human readable and must be decrypted before a machine can perform any computation or analysis. See Table 3.1 for examples of this process.

Table 3.1: *Examples of plaintext messages encrypted with two different cryptographic algorithms*

Plaintext	Ciphertext (MD5)	AES Ciphertext (Base-64)
Firefighter	b74ad4852301652bdbe405413f9a4b49	tfUhMshE82IZCpmNIMHacg==
Robot	5d1eca158c00250d9c4c32d947b7c433	MDoWBb58V8a89UNkUYHuHw==
HARMS	4ae8bcd72369803429490559f21a541b	Q1HqHglXVvA0mvdqOrnfMg==

This study looked to answer whether or not communication between authenticated system actors can be secured against man-in-the-middle attacks. As previously discussed in sections 1.5 and 1.6, other types of network attacks were considered outside the scope of this study.

3.1.1 Hypotheses

The hypotheses for this study were the following:

$H_{0,1}$: The confidentiality of messages sent between authenticated system actors in a HARMS-model network cannot be maintained in the event of an adversary capturing communication.

$H_{\alpha,1}$: The confidentiality of messages sent between authenticated system actors in a HARMS-model network can be effectively maintained even in the event of an adversary capturing communication.

$H_{0,\,2}$: A HARMS-model network cannot be secured against any man-in-the-middle attacks by an adversary.

$H_{\alpha,\,2}$: A HARMS-model network can be effectively secured against a man-in-the-middle attack from an adversary.

3.2 Testing Methodology

The two null hypotheses provide the two major goals of secure communication via HARMS: confidentiality and authentication. During the experiment, both factors must be accounted for and protected in order to successfully reject the null hypotheses. The experiment consisted of two major tests in three scenarios that modeled real-world examples of tasks firefighting robots and HARMS system actors would typically see. In each scenario, an active man-in-the-middle attack was demonstrated prior to securing the HARMS communication. A replay attack was also demonstrated to demonstrate a lesser (but sometimes just as damaging) attack that also requires an adversary in between the sender and recipient. The same attacks were demonstrated after the security had been added as the experiment. The scenarios are as follows.

3.2.1 Scenario 1

In the first scenario, two computer nodes were communicating via HARMS. These computers existed as virtual machines that simulated commands that would typically be seen in a large-scale enterprise environment with sensors for things like temperature, humidity, and other physical building environment parameters. As an example, consider the heating, ventilating, and air conditioning (HVAC) system of a large warehouse or commercial building. These systems typically employ sensors for not only temperature and humidity, as well as smoke and carbon monoxide detectors for fire prevention. When these distributed sensors take readings, they can be communicated in many different ways, including decentralized large-scale networks and supervisory control and data acquisition (SCADA) systems.

In a scenario where various sensor readings are communicated to one or more nodes of a network, it is crucial that data being sent is not tampered with or changed in any way. If a malicious entity were able to gain access to the communications, they could perform a man-in-the-middle attack to change the information being sent, or send new commands that could cause significant damage, such as activating fire sprinklers or turning off air conditioning systems. This scenario is just an example, but has recently become an increased target of cybercriminals. According to an FBI memo, an air conditioning company in New Jersey was compromised when attackers gained access to their incident command system ("Is your HVAC (air conditioning) the next SCADA target?", 2013). Also, although outside the scope of this project's scenario, a consideration of the security of a system like HVAC is important because these systems can often times be used a pivot where the adversary compromises these systems first and then attacks more critical systems afterward, as was the case in the famous Target data breach in November 2013 (Krebs, 2014).

Therefore, for this experiment, the first scenario involved computers communicating via HARMS. Their communications simulated sensor output being sent back to a command and control server. A man-in-the-middle attack was demonstrated to change the content of what was being sent. As an example, machine A sent "Temperature: 85 degrees F" to machine B, simulating a warm environment of a server room that requires air conditioning to be activated. However, the man-in-the-middle attack changed the contents to read "Temperature: 65 degrees F", which means that machine B did not turn on the air conditioning. This is a simulation of an attack that could cause failure of the machines in that server room due to overheating. A replay attack was also demonstrated by capturing the plaintext contents of the message and sending them an arbitrary amount of times. The secure communication was added, and the man-in-the-middle and replay attacks were run again to demonstrate they are longer feasible.

3.2.2 Scenario 2

The second scenario was also modeled on a real-world situation at the heart of firefighting robot operation. Firefighting robots can be issued commands via a human-operated remote control or from one or more system actors of a HARMS network. In either case, a command on how the robot should move or what task should be accomplished was sent. Again, the security of these commands is vital to the operation of the firefighting robots and the mission, because if the commands were altered, the fire could spread and cause more damage or loss of life. In this scenario, a ground vehicle robot was issued commands via HARMS for basic movement, such as moving forward, turning left or right, etc. A man-in-the-middle attack was demonstrated as in the first scenario to change the command such that the robot moved in an unpredictable and unwanted way. A replay attack was also demonstrated to arbitrarily move the robot in undesirable ways without needing communication from an authorized actor. Again, secure communication was added, and the attacks were repeated to demonstrate they are no longer feasible.

3.2.3 Scenario 3

The final scenario demonstrated the ability to use multicast and broadcast messages in a HARMS network. In the previous two scenarios, messages were sent as unicast, which means that there was one sender and exactly one recipient. Multicast messages are intended for multiple recipients, which the sender can specify. Broadcast messages are sent to all known peers of the HARMS network. The third scenario combined the machines used in the previous two scenarios and focused on messages that are sent via multicast and broadcast. This scenario was meant to indicate that the proposed solution is generalizable to multiple pairs of

HARMS actors. Like before, a man-in-the-middle attack and a replay attack were both run on the unsecured communication. After the secure communication, the attacks were attempted again, although this time they were unsuccessful.

3.2.4 Implications

The man-in-the-middle attack used in all three scenarios tests the capabilities of both confidentiality and authentication of the HARMS network. The property of confidentiality, keeping information secret from entities that do not have authority to see it, is important to maintain in communication because there is no inherent safeguards preventing an adversary from capturing wireless communication and analyzing or using it. If an adversary can see and understand communication sent, he or she can change it in unexpected ways. Furthermore, the ability to change a message indicates a lack authentication; that is, the property of having authority to be a member of the network and effectively being allowed to send and receive communications.

Man-in-the-middle attacks can be performed with varying degrees of severity. As seen in the MiG-in-the-middle example, adversaries were able to capture communication used to authenticate the opponent. They relayed these messages and were able to deceive the system and masquerade as an ally, not a foe. This is also known as a replay attack, but it's important to note that the adversaries had to be actively capturing traffic and relaying it, making it valid only in that moment. They couldn't have simply captured traffic from some event and replayed it at any time to gain access. Simple replay attacks are not as advanced but can be just as successful, meaning these attacks should not be discounted.

In fact, replay attacks, in any capacity, test authentication of the HARMS network even when confidentiality is achieved. Here, if an adversary can send messages that appear to be valid, even when he or she cannot understand the communication being sent, is still a significant vulnerability. In other words, even if the IFF messages in the MiG-in-the-middle example were encrypted, they would still be susceptible to a replay attack without other safeguards in place to allow messages to expire. Another variety of a man-in-the-middle attack involves changing the contents of the message in between the sender and the recipient. This can be done when confidentiality and authentication are not present, and an adversary can change the contents of a message directly while in transit. The man-in-the-middle attacks described here were attempted to the best of their ability in the scenarios described in the previous sections.

3.3 Implemented Solution

The solution to secure the communication within the HARMS system addressed both the issues of confidentiality and authentication. Confidentiality is achieved by implementing cryptographic functions on messages that are sent between peers. Not only are the contents of the messages encrypted, but integrity (and non-repudiation) of the message is provided with a cryptographic hash function, called a keyed-hash message authentication code (HMAC). This requires a secret secret to compute that only authorized users would have. Both the encryption of the message as well as the hash function require symmetric cryptographic keys. Therefore, a key exchange protocol is required to establish these keys for each pair of peers that want to communicate. This was achieved via a form of the Diffie-Hellman Key Agreement Method (Rescorla, 1999).

3.3.1 Diffie-Hellman Key Agreement

Keys for both the cryptographic protocol as well as the message authentication code must be negotiated for all pairs of peers separately. Each time symmetric keys need to be negotiated for a new pair of peers, a Diffie-Hellman Key Agreement (DHKA) process occurs, as described here.

1. Peer 1 generates a prime generator g, a large prime number p, and a private value a.

2. Peer 1 computes $A = g^a \bmod p$ and sends g, p, and A (the result) to Peer 2. It is important to note that Peer 1 keeps a secret.

3. Peer 2 generates its own secret, b, and uses g and p to compute $B = g^b \bmod p$.

4. Peer 2 sends B to Peer 1 but keeps b secret.

5. Peer 2 computes the shared secret $Z = g^{ab} \bmod p$ by calculating $A^b \bmod p$.

6. Peer 1 meanwhile computes the shared secret $Z = g^{ab} \bmod p$ by calculating $B^a \bmod p$.

After the Diffie-Hellman Key Agreement concludes for this pair of peers, they each now have an identical shared secret, unique to that pair of peers. Peer 1 and Peer 2 both take the shared secret and feed it as input into an HMAC-based Key Derivation Function (HKDF), deriving from it an arbitrary number of cryptographically-strong secret keys (Krawczyk & Eronen, 2010). Both Peer 1 and Peer 2 use the same shared secret from the DHKA into as well as identical (and optional) other parameters to the function known as the "info" and the "salt". Afterward, Peer 1 and Peer 2 now have identical cryptographically-strong keys for both encryption and message authentication.

There are several things that are important to note. The inherent security of the shared secret negotiated after the Diffie-Hellman Key Agreement relies on the hardness of the Decision Diffie-Hellman (DDH) problem (Boneh, 1998). This assumes that no efficient algorithm can distinguish between $\langle g^a, g^b, g^{ab} \rangle$ and $\langle g^a, g^b, g^c \rangle$. In other words, given the values g^a, g^b, and g^c, it is computationally infeasible to determine whether or not $g^c = g^{ab}$. A related problem is the Computational Diffie-Hellman (CDH) problem, which is given g^a and g^b, it is computationally infeasible to calculate g^{ab}. Both of these problems provide assurance that an adversary cannot eavesdrop on a Diffie-Hellman Key Agreement session and learn the shared secret.

In this solution, each pair of peers need to first negotiate cryptographic keys via DHKA before they can begin to communicate with each other. The keys for encrypted messages as well as message authentication are valid for that session; the pair of peers can use them while they communicate, but whenever the program is run in future scenarios, each pair must re-negotiate keys via DHKA. This leads to a desirable property of the cryptographic keys. Furthermore, in this solution, the negotiated keys are known as ephemeral, in that they are only used for a specific session and not written to a file or used again for future sessions.

3.3.2 Message Encryption and Authentication

After session keys are negotiated via a Diffie-Hellman Key Agreement process (described in the previous subsection), a pair of peers can now utilize strong cryptographic protocols to provide confidentiality and authentication. Messages are transformed in the following way:

1. Peer 1 wants to send a message M to Peer 2.

2. Peer 1 computes the HMAC of M using the HMAC session key negotiated for Peer 2 (the hash algorithm used in this solution is SHA-256).

3. Peer 1 concatenates HMAC(M) with M itself.

4. Peer 1 encrypts the concatenation with AES (128-bit key size, CBC mode of operation, PKCS #5 Padding) i.e., $AES_{DH}(HMAC_{DH}(M) + M)$.

5. Peer 2 receives the ciphertext and decrypts using the AES session key negotiated for Peer 1.

6. Peer 2 computes the HMAC of M using the HMAC session key negotiated for Peer 1.

7. If the computed HMAC matches the HMAC provided in the plaintext of the transmission, Peer 2 accepts the message from Peer 1.

Message authentication relies on two factors. First, an adversary would need to compromise the encryption of the transmission to access the message and the HMAC output in the first place. This is assumed a very difficult problem, as no attacks are known to break the AES algorithm when implemented correctly other than a fully exhaustive search, which is considered computationally infeasible (Biryukov, Dunkelman, Keller, Khovratovich, & Shamir, 2009). Second, even if the encryption can be broken by an adversary who wanted to change the contents of the message, the adversary would have to recompute the HMAC using the session key (which he or she does not have access to, as discussed in the previous subsection) or brute force an appropriate HMAC such that the message is validated. Collision resistance, the property where a certain hash output can be generated for a given plaintext input without actually computing the hash (i.e., without having the key) is strong for SHA-256, this solution's underlying hash function for the HMAC. In fact, no known collisions exist for the SHA-2 family (Schneier, 2012). Note that because session keys are negotiated for each pair of peers, even authenticated members of the HARMS network could not tamper with the authentication of messages unless they were the sender or recipient of the message itself.

3.3.3 Protections Against Man-in-the-Middle Attacks

The previous two subsections offer confidentiality and message authentication for communication in a HARMS network. Man-in-the-middle attacks by adversaries or even HARMS users that are not directly involved in the transmission of the message are no longer possible with the message itself. However, two attack vectors still exist. First, adversaries can perform a man-in-the-middle attack on the Diffie-Hellman Key Agreement process itself (see Figure 3.1). Here, the adversary exists in between the two peers during DHKA and effectively negotiates a shared secret for each. A sophisticated man-in-the-middle attack would decrypt the contents from one peer and re-encrypt them using the other shared secret to forward the message to the originally intended recipient, meaning both parties would be unaware of the attack.

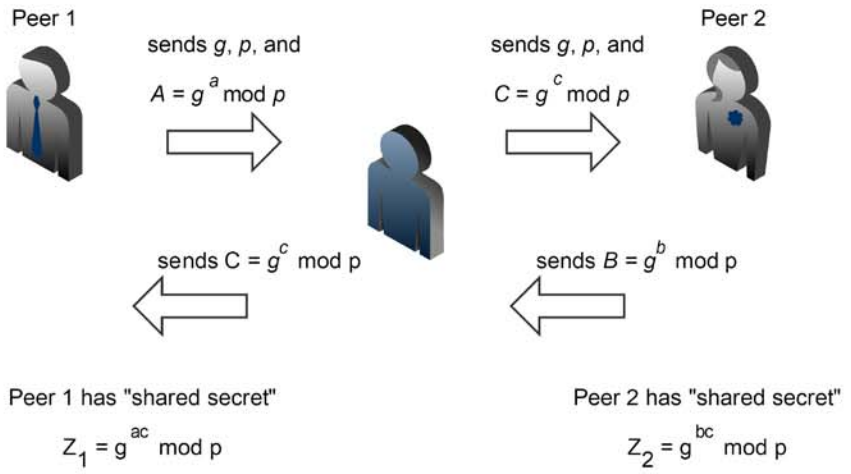

Peer 1
Peer 2
sends g, p, and $A = g^a \bmod p$
sends g, p, and $C = g^c \bmod p$
sends $C = g^c$ mod p
sends $B = g^b$ mod p
Peer 1 has "shared secret" $Z_1 = g^{ac}$ mod p
Peer 2 has "shared secret" $Z_2 = g^{bc}$ mod p

Figure 3.1.: A man-in-the-middle attack on the Diffie-Hellman Key Agreement

Traditionally, this attack is mitigated in real-world scenarios with the use of public key infrastructure (PKI). This is a system involving the use of a trusted certificate authority (CA) who affirms the identity of users or entities on the Internet. However, this is a centralized approach that relies on a CA server to cryptographically sign information relating to the digital identities of users that want to communicate. Because HARMS is inherently a decentralized network, a slightly modified solution is implemented. Rather than signing identities with a trusted authority, all communication, including DHKA, is encrypted with a pre-shared key. This secret key must be installed on all members who wish to participate in a HARMS network. Then, due to the complexity of defeating the AES algorithm as previously discussed, successful decryption of the DHKA communication validates the identity of a HARMS system actor or peer. This idea is similar to the notion of pre-shared keys used in the Internet Key Exchange (IKE) protocol found in IPSec (Kaufman, Hoffman, Nir, Eronen, & Kivinen, 2014). This cryptosystem not only provides ephemeral session keys, as discussed in Section 3.3.1, but it also has the property of forward secrecy, in that session keys cannot be compromised even in the event that the pre-shared key is compromised. In other words, if an adversary captures messages encrypted using this solution and also cracks or otherwise discovers the pre-shared key, he or she still cannot decrypt the contents of the messages, as the Diffie-Hellman Key Agreement process does not disclose anything that an adversary can use to construct the session keys.

The second attack vector to consider is the concept of a middleperson capturing traffic to be able to relay to another node or replay in some way. For example, even when a message is properly encrypted and authenticated between a sender and recipient, if the session is still active, then the session keys are still valid. An adversary who captures that traffic and replays it during that session will force the recipient to verify the message a second time. To provide a real-world scenario for this attack vector, consider a remote-controlled electric lock for a door. A cryptographically-secured message is sent to the door to unlock and open, and the door shuts automatically shortly thereafter. If an adversary captured the command, although encrypted, to the door control, the adversary could still open the door by simply replaying the encrypted message.

In order to protect against this very simple attack, a timestamp is added to the message. As long as the message has occurred within an acceptable threshold of time and the HMAC is verified, the message is considered valid. For the purposes of this solution, four seconds is considered an acceptable threshold (see Section 4.4 for details). Note that this solution requires synchronized time between all HARMS actors to maintain functionality.

The following steps summarize the full solution implementation. Note that “PSK” stands for pre-shared key, and “E” denotes encryption via AES with a 128-bit key in CBC mode of operation with PKCS #5 Padding (Kaliski, 2000), when appropriate. "D" denotes decryption of AES with the same parameters, and "H" denotes computing the mac output using HMAC-SHA-256. Finally, "DH_A" and "DH_H" denote the session keys derived for AES encryption and HMAC computation, negotiated via Diffie-Hellman Key Agreement.

1. Peer 1 sends Diffie Hellman public parameters $E_{PSK}(g)$, $E_{PSK}(p)$, and $E_{PSK}(g^a \text{ mod } p)$ to Peer 2. Peer 1 keeps a private.

2. Peer 2 decrypts $D_{PSK}(E_{PSK}(g))$ to get g, $D_{PSK}(E_{PSK}(p))$ to get p, and $D_{PSK}(E_{PSK}(g^a \text{ mod } p))$ to get g^a mod p.

3. Once Peer 2 decrypts the public parameters, they are used to compute g^b mod p. Peer 2 keeps b private.

4. Peer 2 sends $E_{PSK}(g^b \text{ mod } p)$.

5. Peer 1 computes $D_{PSK}(E_{PSK}(g^b \text{ mod } p))$ to get g^b mod p.

6. Both peers compute the shared secret $Z = g^{ab}$ mod p.

7. Both peers feed Z into the HKDF to obtain DH_A and DH_H, valid for that session only.

8. Peer 1 wants to send a message to Peer 2. Peer 1 first appends the current time to the message, $M = msg + time$.

9. Peer 1 computes $H_{DH_H}(M)$.

10. Peer 1 sends $E_{PSK}(E_{DH_A}(H_{DH_H}(M) + M))$ to Peer 2.

11. Peer 2 computes $D_{PSK}(E_{PSK}(E_{DH_A}(H_{DH_H}(M) + M)))$ to get $E_{DH_A}(H_{DH_H}(M) + M)$.

12. Peer 2 computes $D_{DH_A}(E_{DH_A}(H_{DH_H}(M) + M))$ to get $H_{DH_H}(M) + M$.

13. Peer 2 takes M and computes $H_{DH_H}(M)$. If it is not identical to what is in step 12, the message is not valid.

14. Peer 2 takes the current time and compares it to *time* inside M. If it not within a configured threshold, the message is not valid.

15. If the message is valid, Peer 2 can perform an action or send a message to Peer 1 using steps 8 through 14 as needed.

3.4 Measure for Success

For each of the three scenarios, an active man-in-the-middle attack was run on HARMS actors communicating in a HARMS network, first without the secure solution. It was expected that in all three scenarios, the man-in-the-middle attack would be successful. This was performed as a proof-of-concept to demonstrate the attack as it would occur in a realistic environment. The man-in-the-middle attack was also ran on the secure solution. The measure for success for this project was to see whether or not the attacks were successful after the secure communication was in place.

The second null hypothesis, relating to authentication, was rejected if a man-in-the-middle attack is considered computationally infeasible. This can be shown by a failure to break the cryptographic algorithms in place by the solution or a failure to perform a replay attack within a certain threshold of time. In other words, $H_{0,2}$, was rejected if attacks were not successful in the amount of time determined in the study. If these attacks were not successful, $H_{\alpha,2}$ was accepted.

3.4.1 Data Collection

For each scenario, messages were sent one hundred times from one HARMS actor to another. As discussed previously, this was done for each scenario twice: once without secure communication, and once again with the secure solution. In all cases, the messages and information about sender and recipient was captured. Also, an adversary attempted to capture traffic and sent responses to the correct HARMS actor to bypass authentication and gain access to communication.

3.4.2 Variables

The independent variable in this study was the message that the robot communicates. After secure communication was implemented, the independent variable existed as the ciphertext.

The dependent variable in this study was the ability to read the message. Furthermore, the dependent variable was the ability to capture a message.

3.5 Instrumentation

The technology used for all HARMS actors was a Java application, providing an interface for system actors to add peers, send and receive unicast, multicast, or broadcast data, and keep track of previously-sent messages. All scenarios utilized HARMS in this Java application in both the original and secure versions.

To demonstrate the network attacks, another machine utilized standard traffic sniffing and capture tools such as Wireshark and tcpdump. Man-in-the-middle attacks were done via Ettercap. Replay attacks were demonstrated by taking the raw contents of the captured packets and sending them via netcat or in a Python script.

3.6 Threats

Interference or noise during wireless transmission were seen as a threat to this experiment. If the experiments were performed with other wireless devices nearby, there could have been interference or a significant change in the signal-to-noise ratio. Other threats included side-channel attacks and denial of service attacks, which would have rendered communication unusable but were outside the scope of this project. Finally, because the experiments are a proof-of-concept using robots that are similar but not identical to the firefighting robots, changes might be need to occur when adapting the secure communication developed here for use with the firefighting robots.

3.7 Summary

This chapter provided the framework and methodology used in this study. First, two major goals of securing communication were identified: confidentiality and authentication. These set up two hypotheses that can be tested with an experiment involving three different scenarios. In each scenario, both a man-in-the-middle attack as well as a replay attack were performed on the original HARMS system and the secure solution implementation.

The first scenario involved two virtual machines communicating with each other, simulating a command and control environment with a temperature sensor. The second scenario used a vehicular robot and a machine that provided commands to move the robot in specified ways. The third scenario combined these machines to test the multicast and broadcast capabilities of the HARMS system.

The secure solution was presented in this chapter, and it addresses the security goals laid out previously. Confidentiality is achieved using well-known, standard cryptographic methods. Non-repudiation and message integrity are also provided, making the solution robust to many different attacks. Pre-shared keys provide decentralized authentication in this system by indicating knowledge only true system actors could have, a method of identity. Finally, a timestamp is added to the message so that replay attacks against messages using even the most sophisticated encryption is not possible. The next chapter provides details on the results of the experiment performed as described by the methodology and framework of this chapter.

CHAPTER 4. RESULTS

After setting up the framework and methodology for this research in the previous chapter, the experiment was run and data was collected. This chapter discusses the results of the experiment under those conditions. Each section will provide details on the experiment as it applies to each scenario, both with the original implementation of the HARMS system as well as with the secure solution in place. For each scenario, details on the attacks that are performed are given as a proof-of-concept, and then that same attack was tested against the secure communication solution. Each time, the results are summarized to indicate a measure of success. Afterward, the results are analyzed to draw meaningful conclusions based on the hypotheses that were provided in the previous chapter. The following chapter will summarize the complete project and discuss future research avenues.

4.1 Scenario 1

The first scenario (see Table 4.1) involved using two virtual machines as HARMS actors, “Alice” and “Bob”. Alice was simulating a temperature sensor in a server room, and Bob was simulating the HVAC control for air conditioning. If the server room temperature exceeds 82 degrees Fahrenheit, the air conditioning should turn on to start cooling the room. To begin the test, 100 commands were sent from Alice to Bob indicating the temperature of the server room as 85 degrees Fahrenheit. An attacker “Mallory” was listening on the network and began a man-in-the-middle attack.

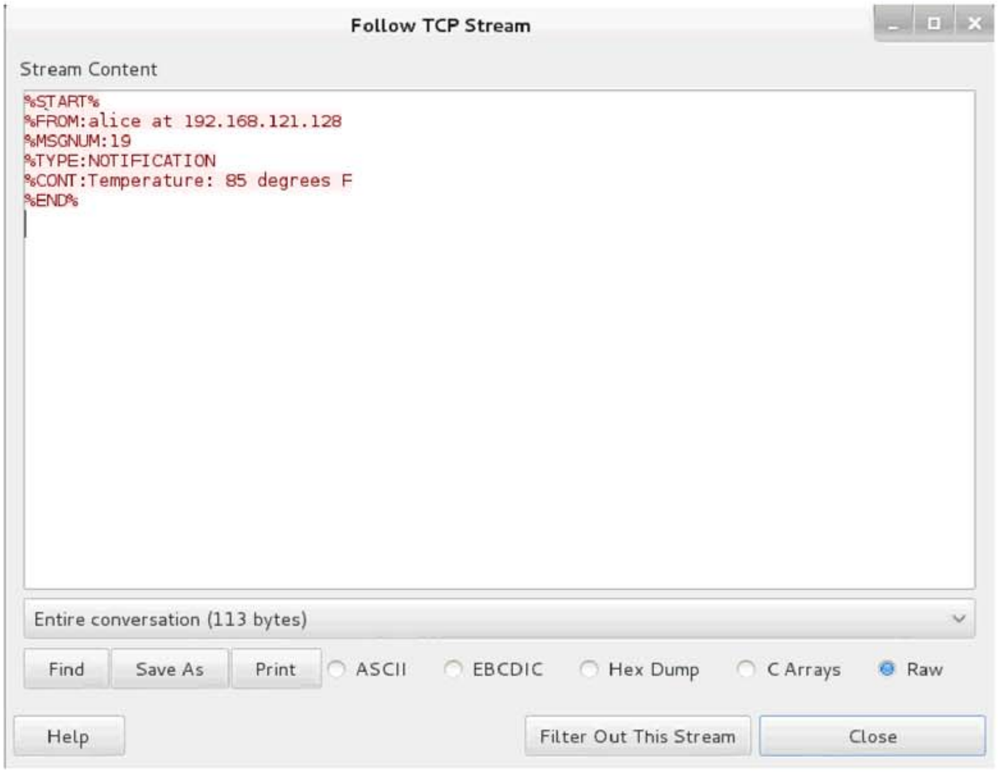
The man-in-the-middle attack used by Mallory was two-fold. First, Mallory listened to the communication sent between the two HARMS actors to first

Table 4.1: *Properties of the two HARMS actors used in the first scenario*

Parameter	Machine 1	Machine 2
Name	alice	bob
IP Address	192.168.121.128	192.168.121.129
Operating System	Linux ubuntu 3.13.0-45-generic	
CPU	2 Processors with 2 cores per processor (4 total cores)	
RAM	2048 MB	
Network Adapter	NAT	
Java Version	1.7.0_75	
VMWare Version	VMWare Workstation 11.1.0 build-2496824	
vmware-tools	9.9.2.44151 (build-2496486)	
Host	Windows 8.1 Pro, 64-bit (Build 9600) 6.3.9600	

understand the message protocol. Figure 4.1 shows the message from a packet capture using Wireshark. Mallory could see the contents of the message in plaintext (i.e., unencrypted) and understood the organization of the message (i.e., how to format the contents). Mallory began to change the contents of the message that Bob saw such that the temperature of the server room was no longer reported correctly. Specifically, Mallory used the exploitation tool Ettercap to write an etterfilter for this attack. The filter was as follows:

```
if(ip.proto == TCP && tcp.dst == 8888){
  if(search(DATA.data, "%CONT:Temperature: 85 degrees F")) {
    replace("%CONT:Temperature: 85 degrees F",
         "%CONT:Temperature: 65 degrees F");
    msg("Filter has run. Detected 85 degrees");
  }
}
```



```
%START%
%FROM:alice at 192.168.121.128
%MSGNUM:19
%TYPE:NOTIFICATION
%CONT:Temperature: 85 degrees F
%END%
```

Entire conversation (113 bytes)

○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

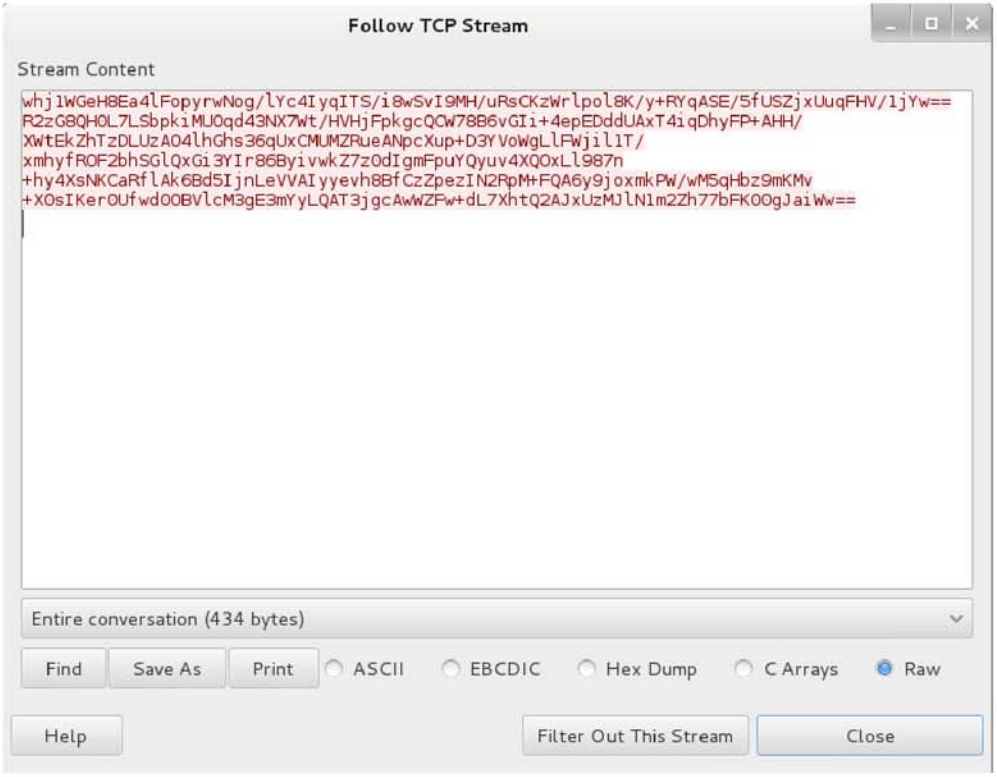
Figure 4.1.: A packet capture of an unencrypted HARMS message

Once the filter was applied and an ARP spoof attack had been performed to have the data sent to Mallory's machine, she could successfully modify the contents of the message and then send them on to Bob. HARMS actors Alice and Bob were not aware of the attack; Alice believed "85 degrees F" was being sent to Bob. Bob was receiving messages indicating the temperature was only 65 degrees, however. Here is a snippet of the output from Bob's machine:

```
[java] From: alice at 192.168.121.128
[java] Message number: 35
[java] Type: NOTIFICATION
[java] Contents:
[java] Temperature: 65 degrees F
```

The messages appeared to be coming from Alice, so Bob took no action in turning on the air conditioning.

Mallory had another variant of the attack to test. Mallory also captured a message that Alice sends and replayed it to Bob. Although it appeared to come from Alice, Alice had no intention of sending more messages. Mallory takes the packet capture from Wireshark and saved the contents of the stream into a file. She used netcat to send that to Bob, who accepts it without issue. Table 4.2 summarizes the results for the attacks of this scenario.



```
whj1WGeH8Ea4lFopyrwNog/lYc4IyqITS/i8wSvI9MH/uRsCKzWrlpol8K/y+RYqASE/5fUSZjxUuqFHV/1jYw==
R2zG8QH0L7LSbpkiMU0qd43NX7Wt/HVHjFpkgcQCW78B6vGIi+4epEDddUAxT4iqDhyFP+AHH/
XWtEkZhTzDLUzA04lhGhs36qUxCMUMZRueANpcXup+D3YVoWgLlFWjil1T/
xmhyfROF2bhSGlQxGi3YIr86ByivwkZ7z0dIgmFpuYQyuv4XQOxLl987n
+hy4XsNKCaRflAk6Bd5IjnLeVVAIyyevh8BfCzZpezIN2RpM+FQA6y9joxmkPW/wM5qHbz9mKMv
+XOsIKer0UfwdO0BVlcM3gE3mYyLQAT3jgcAwWZFw+dL7XhtQ2AJxUzMJlN1m2Zh77bFK0OgJaiWw==
```

Entire conversation (434 bytes)

○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

Figure 4.2.: A packet capture of the encrypted version of a HARMS message

Now that the proof-of-concept for both the active man-in-the-middle and simple replay attack have been demonstrated, the secure communication solution was put in place. Alice and Bob, who have the pre-shared key installed on their machines, communicated as normal. Mallory attempted the same man-in-the-middle attack with Ettercap, but this time it was unsuccessful in all attempts. This was due to the fact that Mallory did not see the same information anymore when capturing the traffic. Figure 4.2 indicates that Mallory could only see encrypted information.

In order to perform this man-in-the-middle attack, Mallory would need to decrypt the contents of the message in order to replace certain words. As discussed in Section 3.3.3, this requires removing two layers of AES encryption to get to the message: one using the pre-shared key, and the other using the Diffie-Hellman session key. Furthermore, Mallory would have to craft an HMAC of the message once it is changed. This method can only be done via exhaustive key search, which is computationally infeasible.

Table 4.2: *The summary of the two network attacks demonstrated on the original HARMS system (denoted as harms) as well as the implementation with secure communication (denoted as harms-secure) for the first scenario*

	harms	harms-secure
MitM Vulnerable	Yes	No
No. of attempted attacks	100	100
No. of successful attacks	100 (100%)	0 (0%)
Replay Vulnerable	Yes	No
No. of attempted attacks	100	100
No. of successful attacks	100 (100%)	0* (0%)

Mallory was forced to cut her loses and attempt only the simple replay attack instead. She had captured the encrypted traffic as shown in Figure 4.2, so she could replay this as she did with the unencrypted version. This failed to work, however, due to the protections in place using current system time. Table 4.2 summarizes the results for Scenario 1. Please note, the asterisk placed near the 0% effectiveness of the replay attack for the secure HARMS system indicates that a replay attack is still theoretically possible if the adversary can beat the timing threshold set in the program.

4.2 Scenario 2

The second scenario was similar in process to the previous one, but the two machines used were quite different. This was done intentionally to ensure that HARMS communication and the secure solution work on a variety of machines. Table 4.3 includes the details of these HARMS actors. Like in the first scenario, both a man-in-the-middle attack was demonstrated as well as a replay attack, using traffic that an adversary (Mallory) captured. In this scenario, the replay attack is actually more beneficial, as it is easier to control the movements of a robot with arbitrary commands than it is to wait for commands to come from a legitimate sender to change.

Table 4.3: *Properties of the two HARMS actors used in the second scenario*

Parameter	Machine 1	Machine 2
Name	micro	rasp
IP Address	192.168.1.16	192.168.1.24
Operating System	Windows 8.1 Pro 64-bit	Linux raspberrypi 3.12.22+
CPU	ARM1176JZF-S (700 MHz)	Intel Core i5-3570K
RAM	16 GB	512 MB
Network Adapter	Broadcom BCM57781	Edimax EW-7811Un
Java Version	1.8.0_31	1.7.0_40

To start, the machine “micro” sent commands to the robot “rasp” to demonstrate basic movement. These commands included moving forward and backward and moving left and right. These commands are very similar to how firefighting robots would be commanded for navigation or locomotion in a real-world scenario in a HARMS network. Because these commands were sent via wireless (IEEE 802.11 WiFi), Mallory could easily listen and capture traffic being sent to the robot. For each of the four commands, 25 messages were sent to the robot instructing it to move in that way. Mallory was able to perform the man-in-the-middle attack via Ettercap, very similar to how it was achieved in the previous scenario. The etterfilter used was:

```
if(ip.proto == TCP && tcp.dst == 8888){
  if(search(DATA.data, "%CONT:go forward")) {
    replace("%CONT:go forward", "%CONT:turn left);
    msg("Filter has run. Detected going forward");
  }
}
```

This etterfilter included checks for all four commands (forward, backward, left, and right) sent to the robot and changed them to be a undesired command. Again, in all 100 messages, the HARMS actors and network was susceptible to the attack.

Mallory then used a captured packet to replay commands to the robot in an undesired manner. The robot saw that the packet stated it was from a known peer “micro”, so it accepted the message as is. Table 4.4 summarizes the results of these two attacks for the second scenario.

The secure communication solution was then put in place. Again, each authorized actor needed the pre-shared key installed in order to communicate properly. Once the pre-shared key was in place for both micro and rasp, the machines communicated normally. Like in the previous scenario, Mallory was able to view the contents of the messages, but they were encrypted and not discernible without first decrypting the contents. Furthermore, the encrypted contents prevented Mallory from changing the contents, so the man-in-the-middle attack was not successful. Mallory also failed to perform the replay attack, as she could not capture traffic and replay it fast enough for the system to consider the message valid. Again, the asterisk in Table 4.4 indicates that in another version of this experiment, an adversary could be successful if they were quick enough.

Table 4.4: *The summary of both network attacks demonstrated on the original HARMS system (harms) as well as the implementation with secure communication (harms-secure) for the second scenario*

		No. of successful attacks	
	Command	harms	harms-secure
Man-in-the-middle attacks	move forward	25 (100%)	0 (0%)
	move backward	25 (100%)	0 (0%)
	turn left	25 (100%)	0 (0%)
	turn right	25 (100%)	0 (0%)
No. of total attempted attacks		100	100
Replay attacks	move forward	25 (100%)	0* (0%)
	move backward	25 (100%)	0* (0%)
	turn left	25 (100%)	0* (0%)
	turn right	25 (100%)	0* (0%)
No. of total attempted attacks		100	100

4.3 Scenario 3

The last scenario used multiple machines of different makes, capabilities, and architectures to represent a true multiagent network. The HARMS actors used here, summarized in Table 4.5, were mostly actors used in the previous two scenarios with one new machine added. It is important to remember that the HARMS system allows for each actor to maintain their own list of known peers; not all actors must know and communicate will all peers on the network. As such, Table 4.5 also indicates the peer list for each system actor, enumerating each peer a specific actor could communicate with.

Table 4.5: *Properties of the HARMS actors used in the third scenario*

Machine	Platform	Peers list	Previously seen
micro	Windows PC	alice, bob, rasp, apple	Scenario 2
alice	Ubuntu (VM)	bob, micro	Scenario 1
bob	Ubuntu (VM)	alice, micro	Scenario 1
rasp	Raspberry Pi	micro, apple	Scenario 2
apple	Macbook Pro	micro, rasp	New

The previous two scenarios tested the potential vulnerabilities of unicast messages. The third scenario, however, involved sending both multicast and broadcast messages. Multicast messages are sent from one sender to multiple specified peers in that actor's peer list. Broadcast messages are sent to all peers in a particular actor's list. Each type of communication mechanism has potential security implications, so the goal of this scenario was to have the adversary Mallory attempt the previously-used network attacks on both multicast and broadcast messages.

After sending messages in both multicast and broadcast modes, Mallory discovered that these messages appear identical to unicast messages in terms of a packet capture, but sent to multiple actors in quick succession. This meant that the man-in-the-middle attack via Ettercap worked in the same way; a filter was applied, and the contents of the message were changed when they were detected in the contents of the message. If the same message was sent to multiple actors, it was changed similarly for each actor. The replay attack also worked identically to previous scenarios as well. The adversary could arbitrarily send a properly-crafted message to any HARMS actor, as long as that actor was aware of the peer listed in the message as the sender.

Because multicast and broadcast messages were sent as multiple unicast messages, the secure communication implementation was equally successful on these modes of communication as it was in previous scenarios. Each pair of peers negotiated session keys via DHKA, so the message was sent multiple times, encrypted with the same pre-shared key each time but unique session keys per recipient. This solution provided confidentiality to prevent the active man-in-the-middle attack from successfully changing message contents. For the replay attack, one of two events could occur. If the message was captured for a particular HARMS actor and sent to a different one, the message was immediately discarded as incorrect, because it was not decrypted with the appropriate session key. If the message was a replay of something that actor was sent previously, it was decrypted successfully but then marked as invalid, due to the expiration of the timestamp. Table 4.6 summarizes the results of the third scenario.

Table 4.6: *The summary of the two network attacks demonstrated on both multicast and broadcast messages in the original HARMS system (harms) and the implementation with secure communication (harms-secure) for the third scenario*

		No. of successful attacks	
	Message type	harms	harms-secure
Man-in-the-middle attacks	multicast	50 (100%)	0 (0%)
	broadcast	50 (100%)	0 (0%)
No. of total attempted attacks		100	100
Replay attacks	multicast	50 (100%)	0* (0%)
	broadcast	50 (100%)	0* (0%)
No. of total attempted attacks		100	100

4.4 Analysis

After testing the secure communication solution, the results indicate that it was indeed successful in providing confidentiality of message contents and authentication of HARMS actors. Because confidentiality could be maintained even when an adversary captured network communication, the first null hypothesis, $H_{0,1}$, was rejected. Other than application issues with the HARMS system preventing communication from occurring, the results provide evidence to suggest that the solution is robust against man-in-the-middle and replay network attacks without a heavy burden placed on the user to configure complicated system properties. Therefore, the second null hypothesis, $H_{0,2}$, is also rejected. Application issues or extenuating circumstances with the network that cause HARMS communication to fail would affect both the original implementation as well as the secure version, making these issues unrelated to the key exchange or encrypted communication elements of the solution.

Choosing both active man-in-the-middle attacks that altered the contents of the messages on the fly as well as a variant of the attack that allowed a middleperson to capture and replay valid traffic turned out to provide two useful real-world attacks for benchmarking purposes. In an environment modeled in scenario one, where a machine periodically indicates system status via messages to other HARMS actors, a man-in-the-middle attack was most appropriate. On the other hand, when controlling a robot via remote-control commands, a replay attack was most appropriate because the adversary can capture a single valid message and control the robot arbitrary by continuously replaying it, rather than waiting for a HARMS actor to again communicate with it. Both attacks were considered in all three scenarios for comprehensiveness.

After the solution demonstrated resilience toward the replay attack due to verifying system time, it became clear how important synchronized time was between authorized HARMS actors. If two peers who use the pre-shared key communicate (i.e., no malicious activity is actually occurring) but their system clocks are off, the recipient will not validate the message that otherwise would have been acceptable. However, because the threshold for validating a message with timing differences is configurable, an administrator of a HARMS network can change this amount of time for any number of valid reasons (e.g., large geographic distance between actors who are communicating, or low computational resources to perform the encryption and decryption functions).

In considering the system overhead introduced due to the cryptographic functions implemented for the secure communication, a simple test was performed to calculate how much longer a system needed to send a message securely. Using the Raspberry Pi HARMS actor, the least powerful of the machines in terms of computation, small modifications were added to the HARMS application to determine how quickly a full DHKA process could take place and a unicast message could be sent to a peer. Since this application is written in Java, the following code placed before and after the solution will provide an accurate view of the time overhead:

```
ThreadMXBean threadBean = ManagementFactory.getThreadMXBean();
long time = threadBean.getCurrentThreadCpuTime();
```

In the worst case, where the Raspberry Pi system actor must negotiate session keys as well as send a message to a peer, the entire process takes around 3 seconds. Although this is a long time, subsequent messages sent to the robot only take 0.002 seconds on average to process. However, due to this time requirement with the Raspberry Pi, the system-wide threshold for accepting messages is set to 4 seconds. With machines with significantly more power, the timing overhead for DHKA is not noticeable. If a HARMS network is composed of only these machines, this threshold can be reduced significantly.

Finally, after running the experiment for multicast and broadcast messages (and during the implementation of the secure communication), it was discovered that these messages are sent as multiple unicast messages. This indicates an avenue for improvement in future research; this system can be written such that n messages do not need to be sent to n actors via group key management or other cryptographic methods outside the scope of this research.

The next chapter provides a summary of the research as a whole, offers conclusions, and expands upon potential future work.

CHAPTER 5. SUMMARY AND FUTURE WORK

This work was largely inspired by identifying a gap in secure communication from previous work with the HARMS system. Because the HARMS model allows for any number of heterogeneous entities to join together to achieve a common goal, many distinct applications were identified early on, which included the adoption of the HARMS system to the domain of firefighting robots. Due to the critical and sometimes life-threatening nature of firefighting, implementing a solution for secure communication was a prime focus, as no known solutions for a HARMS-model network were present. During the literature review, it became apparent that cyberphysical attacks on critical infrastructure are on the rise, further increasing demand for the addition of security in communications.

Man-in-the-middle attacks were chosen as the scope for this project due to their widespread use, relatively ease of execution, and effective results when performed accordingly. Three scenarios were chosen as the basis for the experiment to provide both a variety in machines performing as HARMS actors and a variety of communication methods. In all three scenarios, two forms of a middleperson attack were demonstrated as a proof-of-concept on the original HARMS system: the active man-in-the-middle attack that changed the contents of the messages in transit, and the replay attack that captured messages and sent them to HARMS actors arbitrarily and repeatedly.

After the attacks were demonstrated, the proposed solution was put in place. This solution provided confidentiality by implementing standard cryptographic methods and key agreement protocols to negotiate ephemeral session keys. Message authentication, integrity, and non-repudiation were also provided, making the solution robust to many different attacks. Pre-shared keys provided decentralized authentication in this system by indicating knowledge only true system actors could have, a method of identity. Finally, a timestamp was added to the message so that replay attacks against messages using even the most sophisticated encryption were no longer possible. These secure communication mechanisms achieved forward secrecy, so even if the pre-shared keys were compromised, no captured traffic could be decrypted meaningfully.

The solution performed well in all tests, as man-in-the-middle attacks and replay attacks were no longer possible against the new system in all three scenarios. Certain concessions had to made, including installing pre-shared keys on all HARMS machines, adding overhead during the key agreement protocol due to some expensive cryptographic operations, and requiring synchronized system clocks for message verification.

5.1 Future Work

Future research could focus on several different paths. First, there are other types of network attacks to consider, such as denial of service attacks. These attacks, which could include wireless traffic jamming, exploitation of the application itself to tax the system or crash it all together, or tampering with the pre-shared keys to prevent communication from succeeding.

The efficiency of the cryptosystem implemented in this research could also be improved so that devices with low computing power can still communicate effectively as HARMS actors. This might include porting the HARMS system into a lower level programming language, such as C.

Although timestamps were introduced as a security mechanism to help prevent replay attacks, other types of checks could also be implemented to further increase security of communication. Global positioning could be used to provide geographic or localization data in the message, to better help prevent a situation similar to the MiG-in-the-middle attack, where actors are very far away from each other in space.

As mentioned in Section 4.4, the implementation of multicast and broadcast messages also has room for improvement. This problem introduces the concept of shared secrets or some form of group key management, a new cryptographic hurdle to overcome. However, if solved, this could lead to improvements in overall performance and cut down on total network traffic.

Finally, applying this research to a real-world scenario that requires firefighting robots would be of particular interest, now that this proof-of-concept has been put in place.

LIST OF REFERENCES

LIST OF REFERENCES

Amano, H. (2002). Present status and problems of fire fighting robots. In *SICE 2002. proceedings of the 41st SICE annual conference* (Vol. 2, pp. 880–885).

Anderson, R. (2008). *Security engineering: A guide to building dependable distributed systems* (Second ed.). Indianapolis, IN: Wiley Publishing, Inc.

Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., & Shamir, A. (2009). *Key recovery attacks of practical complexity on AES variants with up to 10 rounds.* Cryptology ePrint Archive, Report 2009/374.

Bless, E. (2006). Anna konda: the firefighting snakebot. *Engadget.* Retrieved from www.engadget.com/2006/07/23/anna-konda-the-firefighting-snakebot

Boneh, D. (1998). The decision Diffie-Hellman problem. In *Algorithmic number theory* (pp. 48–63). Springer.

Bradshaw, A. (1991). The UK security and fire fighting advanced robot project. In *Advanced robotic initiatives in the UK, IEE colloquium on* (pp. 1–4).

Brenner, B. (2012). *ICS-CERT alert: Natural gas pipelines under attack.* Retrieved from www.csoonline.com/article/2135157/critical-infrastructure/ics-cert-alert--natural-gas-pipelines-under-attack.html

Cherry, S. (2011). Sons of stuxnet. *IEEE Spectrum.* Retrieved from spectrum.ieee.org/podcast/telecom/security/sons-of-stuxnet

Ciancamerla, E., Minichino, M., & Palmieri, S. (2013). Modeling cyber attacks on a critical infrastructure scenario. In *Information, intelligence, systems and applications (IISA), 2013 fourth international conference on* (pp. 1–6).

The Clinton administration's policy on critical infrastructure protection: Presidential directive 63 (Tech. Rep.). (1998, May 22). White House. Retrieved from fas.org/irp/offdocs/pdd/pdd-63.htm

Davis, T. (2015). Propane tank truck fire shuts down interstate 20 in Arlington. *NBCDFW.* Retrieved from nbcdfw.com/news/local/18-Wheeler-Fire-Shuts-Down-Interstate-20-in-Arlington-287324761.html

Dubel, W., Gongora, H., Bechtold, K., & Diaz, D. (2003). An autonomous firefighting robot. *Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA.*

Esmaeili, A., Mozayani, N., & Motlagh, M. (2014, Feb). Multi-level holonification of multi-agent networks. In *Intelligent systems (ICIS), 2014 iranian conference on* (p. 1-5).

Goldman, D. (2013). Hacker hits on U.S. power and nuclear targets spiked in 2012. *CNN Money*. Retrieved from money.cnn.com/2013/01/09/technology/security/infrastructure-cyberattacks/

Grant, C. (2014). Creating the research roadmap for smart fire fighting. *Fire Protection Research Foundation.*

Heller, W. (2011). Firefighting robots in japan. *Robotland*. Retrieved from robotland.blogspot.com/2011/03/firefighting-robots-in-japan.html

Higgins, F., Tomlinson, A., & Martin, K. M. (2009). Survey on security challenges for swarm robotics. In *Autonomic and autonomous systems, 2009. ICAS'09. fifth international conference on* (pp. 307–312).

Homeland security presidential directive 7 (Tech. Rep.). (2003, Dec 17). White House. Retrieved from www.dhs.gov/homeland-security-presidential-directive-7

Is your HVAC (air conditioning) the next SCADA target? (2013). *Cyber Defense Magazine*. Retrieved from www.cyberdefensemagazine.com/is-your-hvac-air-conditioning-the-next-scada-target/

Kaliski, B. (2000). *PKCS #5: Password-based cryptography specification version 2.0* (RFC No. 2898). RFC Editor. Internet Requests for Comments. Retrieved from www.rfc-editor.org/rfc/rfc2898.txt

Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., & Kivinen, T. (2014). *Internet key exchange protocol version 2 (IKEv2)* (RFC No. 7296). RFC Editor. Internet Requests for Comments. Retrieved from www.rfc-editor.org/rfc/rfc7296.txt

Kobayashi, A., & Nakamura, K. (1983). Rescue robots for fire hazards. In *Proceedings of the 1983 international conference on advanced robotics* (pp. 91–98).

Krawczyk, H., & Eronen, P. (2010). *HMAC-based extract-and-expand key derivation function (HKDF)* (RFC No. 5869). RFC Editor. Internet Requests for Comments. Retrieved from www.rfc-editor.org/rfc/rfc5869.txt

Krebs, B. (2014). Target hackers broke in via HVAC company. *Krebs on Security*. Retrieved from krebsonsecurity.com/2014/02/target-hackers-broke-in-via-hvac-company/

Kumar, V., Rus, D., & Singh, S. (2004). Robot and sensor networks for first responders. *Pervasive Computing, IEEE*, *3*(4), 24–33.

Lahr, D., Orekhov, V., Lee, B., & Hong, D. (2013). Early developments of a parallelly actuated humanoid, SAFFiR. In *ASME 2013 international design engineering technical conferences and computers and information in engineering conference* (pp. 1–7).

Lewis, J., Matson, E. T., Wei, S., & Min, B.-C. (2013). Implementing HARMS-based indistinguishability in ubiquitous robot organizations. *Robotics and Autonomous Systems*, *61*(11), 1186–1192.

Man-in-the-middle attack. (2014). *OWASP*. Retrieved from www.owasp.org/index.php/Man-in-the-middle_attack

Matson, E. T., & Min, B.-C. (2011). M2M infrastructure to integrate humans, agents and robots into collectives. In *Instrumentation and measurement technology conference (I2MTC), 2011 IEEE* (pp. 1–6).

McDonald, G., Murchu, L. O., Doherty, S., & Chien, E. (2013). Stuxnet 0.5: The missing link. *Symantec Report*. Retrieved from www.symantec.com/connect/blogs/stuxnet-05-missing-link

Min, B.-C., Matson, E. T., Smith, A., & Dietz, J. E. (2014). Using directional antennas as sensors to assist fire-fighting robots in large scale fires. In *Sensors applications symposium (SAS), 2014 IEEE* (pp. 360–365).

MVF-5. (n.d.). *DOK-ING*. Retrieved from dok-ing.hr/products/firefighting/mvf_5

Naghsh, A. M., Gancet, J., Tanoto, A., & Roast, C. (2008). Analysis and design of human-robot swarm interaction in firefighting. In *Robot and human interactive communication, 2008. RO-MAN 2008. the 17th IEEE international symposium on* (pp. 255–260).

National strategy for homeland security (Tech. Rep.). (2002). Office of Homeland Security. Retrieved from www.dhs.gov/sites/default/files/publications/nat-strat-hls-2002.pdf

Nguyen, C. Q., Min, B.-C., Matson, E. T., Smith, A. H., Dietz, J. E., & Kim, D. (2012). Using mobile robots to establish mobile wireless mesh networks and increase network throughput. *International Journal of Distributed Sensor Networks*, *2012*.

Office of infrastructure protection. (2014). *Homeland Security*. Retrieved from www.dhs.gov/office-infrastructure-protection

Plackett, B. (2012). Rescue me, robot: Machines ready for firefighting duty. *Wired Magazine*. Retrieved from www.wired.com/2012/10/fire-fighting-robots

Rescorla, E. (1999). *Diffie-Hellman key agreement method* (RFC No. 2631). RFC Editor. Internet Requests for Comments. Retrieved from www.rfc-editor.org/rfc/rfc2631.txt

Russell, S., & Norvig, P. (2009). *Artificial intelligence: A modern approach* (Third ed.). Upper Saddle River, NJ: Prentice Hall.

Schneier, B. (2008). Aspidistra. *Schneier on Security*. Retrieved from www.schneier.com/blog/archives/2008/11/aspidistra.html

Schneier, B. (2012). When will we see collisions for SHA-1? *Schneier on Security*. Retrieved from www.schneier.com/blog/archives/2012/10/when_will_we_se.html

Schoen, S., & Galperin, E. (2011). Iranian man-in-the-middle attack against google demonstrates dangerous weakness of certificate authorities. *Electronic Frontier Foundation*. Retrieved from www.eff.org/deeplinks/2011/08/iranian-man-middle-attack-against-google

Schumacher, M., McVay, S., & Landes, J. (1999). Pokey the fire-fighting robot. Retrieved from www.ee.nmt.edu/~wedeward/EE382/SP99/group7_fr.pdf

Skoloff, B., & Cone, T. (2013). Firefighters use drones to battle Yosemite rim fire. *Huffington Post*. Retrieved from www.huffingtonpost.com/2013/08/28/drones-yosemite-fire_n_3833528.html

Smith, B. (2014). Are drones the future of firefighting? *Washington Times*. Retrieved from www.washingtontimes.com/news/2014/jul/5/ are-drones-the-future-of-firefighting

Smith, D. L., Petroka, R. P., Yobs, R. L., Lewis, D., & McCarthy, W. (1985). *A mechanical predesign project in robotic fire fighting* (Tech. Rep.). Monterey, California. Naval Postgraduate School.

Thring, M. W. (1963). The domestic revolution. *Journal of the Royal Society of Arts*, 556–572.

Unisys. (2014). Unisys survey reveals nearly 70 percent of critical infrastructure providers have been breached in the past year. *Unisys*. Retrieved from www.unisys.com/offerings/security-solutions/NewsRelease/ Unisys-Survey-Reveals-Critical-Infrastructure-Providers-Breached

Weiss, G. (1999). *Multiagent systems: A modern approach to distributed artificial intelligence*. MIT press.

What is critical infrastructure? (2013). *Homeland Security*. Retrieved from http://www.dhs.gov/what-critical-infrastructure

Zetter, K. (2015). A cyberattack has caused confirmed physical damage for the second time ever. *Wired Magazine*. Retrieved from www.wired.com/2015/01/german-steel-mill-hack-destruction

NOTES

NOTES

Please note that parts of this thesis were included in a publication currently in review to the 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015).