Countering Eavesdropping Attacks in Software-Defined Radio Networks Using a Moving-Target Defense

Isaac J. Cushman
Georgia Southern University
ic00214@georgiasouthern.edu

Rami J. Haddad
Georgia Southern University
rhaddad@georgiasouthern.edu

Lei Chen
Georgia Southern University
lchen@georgiasouthern.edu

Abstract

Every network system is required to maintain a certain level of security for its users. In mission-critical systems, loss or theft of data can become a serious problem for the users involved. It is possible to increase the level of security offered by the network by confusing the attacker. The attacker will generally follow an attack process that starts with eavesdropping the network to discover the system configuration, and then, once a part of the network, will begin sniffing packets for either theft or destructive purposes. This paper presents a new security method that uses moving target defense in an array of software-defined radio networks to split the data packet into multiple pieces and scramble the packet's order. The probability of the eavesdropper stealing data packets is examined for the proposed system and compared to the system without packet scrambling and fragmentation.

Introduction

Moving target defense (MTD) increases the difficulty for an attacker to breach into a network by continuously changing security parameters. The software-defined radio (SDN) will help the network in MTD to protect sensitive data while changing parameters and causing the attacker to lose the trace. Wireless networks have the disadvantage of having a very wide attack surface, due to the dynamically changing size of the network. In an MTD system, the attack surface is defined as the total possible ways that an attacker can breach the system (Yeung, Cho, Morrell, Marchany, & Tront, 2016). Generally, MTDs are deployed in *ad hoc* type networks with many devices and IP addresses, thus having a larger attack surface. Considering the physical cost of operating an MTD is important when determining the level of system security. For software-defined radio networks, the most common technique is frequency hopping. This technique offers the ability to avoid malicious users from jamming or eavesdropping the network by systematically changing the operating frequency in a specific time interval. There are challenges to frequency hopping, most importantly the necessity to request access to occupy more bandwidth at any given time.

MTD has two main categories, network and host based. The major difference is the localization of the controller. As the name implies, a network-based MTD system will have the entirety of the system controlled by a single, centralized entity, whereas a host-based MTD may have multiple host controllers a specific group of nodes (Green, MacFarland, Smestad, & Shue, 2015). Generally, a network-based MTD system is preferred due to the ease of synchronization; however, it may be less secure as the attack will have access to the whole network at one time.

A fundamental property of MTD is the mapping system, which is responsible for determining what users are considered trustworthy based on authentication or a calculated trust value. Secondary properties of network-based MTD are moving, access control, and distinguishability. This system offers potential in countering eavesdropping and intelligent jamming in a network system; however, when it is deployed in a large-scale system, it seems that packet overhead will lead to more significant losses in communication.

The moving property is the characteristics of MTD that make it harder for an attacker to breach the system. It is broken into three sub-properties: unpredictability, vastness, and periodicity. Being unpredictable means that no one in the system should be able to determine the next step other than the machines communicating. The availability of the changing metrics determines the vastness property. This includes either changing IP address, ports, or frequencies; vastness also aids the unpredictability of the system. Periodicity is responsible for maintaining the synchronization between the devices within the system. This parameter keeps a regular changing time interval that should be kept secret between transmitter and receiver.

The access control property is an enforcement policy within the system to authenticate users as they are requesting access. Like the moving property, access control is also divided into sub-properties: uniqueness, availability, and revocability. The uniqueness property uses the mapping system to make sure that each user is guaranteed individual availability in the system dependent on the authorization. The availability property is set in place so that the system can fill each request; generally, this is determined by a capacity limitation within the network configuration.

The revocation property is a set of rules and standards that can remove a user from the system when a series of checks have been met. Revocation can be due to violations or simply the expiration of a time lease in the network; the user would then need to re-request for access.

The last main property that makes up a network-based MTD system is distinguishability, which gives the system the ability to determine trustworthy users within it. The ability to determine trust is typically complex as a non-malicious user may be requesting more information about the mapping system than what is allowed and then become flagged as a malicious user. The typical method of distinguishing is to use identifiers, such as shared keys, to determine trustworthy users.

In software defined networks (SDNs), denial of service (DoS) comes in the form of a signal jammer. The attack will find the frequency that the pair is on and overcrowd the medium so that the receiver cannot make sense of a retrieved package. One of the most common attacks to a network system is a distributed DDoS attack, in part due to the ease of deploying this attack and to the difficulty of determining real traffic from fake traffic. Several mechanisms have been developed to counter a DDoS attack. However, most of these are reactive, meaning they happen after the attack has already occurred (Aydeger, Saputro, Akkaya, & Rahman, 2016). Using a SDN, it is possible to provide proactive defense mechanisms to seemingly random spikes in traffic on a network. The main challenge of an SDN-based network is a crossfire attack, which determines the location of critical link connections in the network by observing the traceroute messages and then targets those links. Another concern in SDNs is eavesdropping in a network system; it is the process of secretly listening to a network and copying the data as they are sent (Ma, Wang, Lei, Xu, Zhang, & Li, 2016).

To prevent theft in the network, frequency hopping is often viewed as a viable countermeasure.This paper proposes a new security method that uses moving target defense in an array of software-defined radio networks to split the data packet into multiple pieces and scramble the order of the fragments. The effectiveness of the proposed model with fragmentation and scrambling is evaluated and compared to a basic frequency hopping system. The remainder of the paper is organized as follows. The related work section highlights the related work and discusses several key components of moving target defense, such as common configurations. The proposed solution section introduces and explains the proposed system design and implementation, while the experimental results section discusses the experimental setup and its results. The paper ends with a conclusion summarizing the results.

Related Work

The network design for moving target defense can be modeled with a varying number of available computing clusters, which the attacker will be trying to access. This array of computing clusters will have a common controller used to keep synchronization during transition periods. This network is visualized in Figure 1.

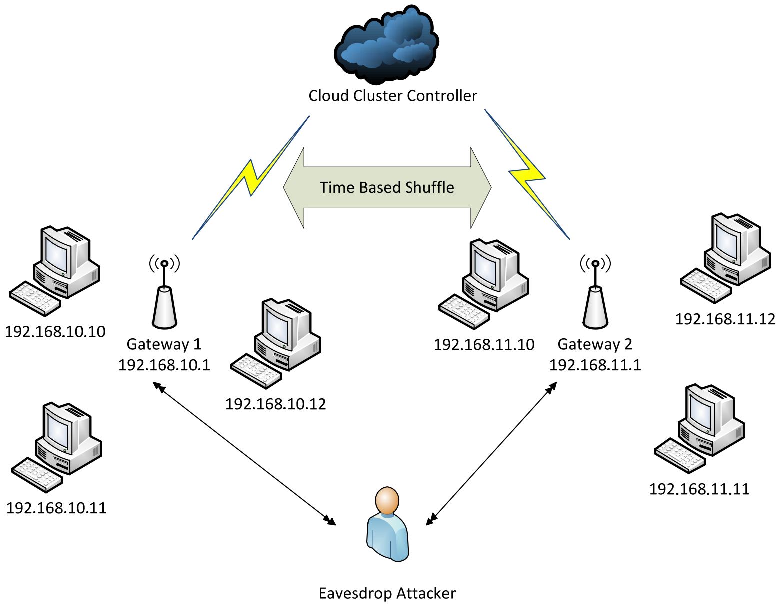

Figure 1. Moving target defense system with attacker.

The connected computers to the gateway represent any variety of applications that can exist on a network (physical or virtual machines, software-defined radios, or mobile devices). The attacker represents any computer system that would attempt connection to any part of the network; once in, the attacker would begin stealing or destroying information. The cloud cluster controller is tasked with keeping a synchronized connection for the entire network by establishing the changing interval, the current IP, port, or frequency, and the next hop configurations.

Moving Target Defense Configurations

Two techniques of an MTD are host-based and network-based. In a network-based MTD, network properties periodically change to increase the difficulty for the attacker to get into the network; altering the IP address is more common (Yeung et al., 2016). Utilizing IPv6 in MTD provides the network with a wide array of possible IP addresses for hopping purposes. The challenge to this technique, however, is the increase in network overhead as hopping addresses generates network discovery protocol and messages. IPv6 provides a substantial advantage over IPv4 in large-scale network systems and services because IPv6 allows for a 128-bit address.

An MTD network using IPv6 commonly referred to as MT6D uses an encapsulation method to confuse the observer by generating a false sense of network activity. Using MT6D proposes a defense against an attacker by causing the attacker to spend a much higher amount of resources on reconnaissance (Yeung et al., 2016). The moving property can be handled by the DNS server with a short time to live assigned value so that IP addresses change frequently. The use of IPv6 is highly sought after in MTD systems because IPv6 offers a large array of varying IP addresses. The DNS server can also handle access control by assigning users to a unique portion of the mapped IP addresses and revoking them when needed. Distinguishability is the most challenging to deploy in a system because of the ability for an attacker to passively access a system (Corbett, Uher, Cook, & Dalton, 2014). Researchers have been investigating the use of ISP DNS systems to provide trustworthiness in a system.

The mechanisms that change within an MTD system are categorized based on the mechanisms and the type of pattern it follows. Three of the mechanisms are software transformations, dynamic platform techniques, and network address shuffling (Cai, Wang, Luo, Li, & Wang, 2016). The idea of software transformations is to focus on the applications that are running on the system. In this case, the software or application will exist in different variants, which will be randomly selected to be the active software version. Dynamic platform techniques involve dynamically changing properties in the operating system and hardware. Recent methods of dynamic platform techniques are to use cloud-based systems to store the operating system variants and load them accordingly. In network address shuffling, the primary goal is to prevent reconnaissance in the system. This is the most common technique in MTD; it involves changing the IP address or port that the network is communicating on.

MTD has three fundamental patterns: hidden, variation and assisted. In the hidden pattern, the attacker can get into the network for a variable amount of time; however, when repeating the reconnaissance stage, the network will have appeared to be no longer active. The variation pattern is comparable to hidden; however, when the attacker makes a second pass, the network will have a different set of security protocols, preventing access.

Moving Target Defense Configurations

The first stage of the attack process on a network is the reconnaissance stage; the focus of this stage is to determine the best angle of attack. It is because of this understanding that defenders of cyber-attacks work to make reconnaissance very difficult. Eavesdropping is one of the most significant challenges to stop malicious acts in a network system; it is the process of secretly listening to a network and copying the data as it is sent (Ma et al., 2016). Traditionally, encryption and authentication are backbone layer defenses that are passive in the network. This means that the encryption system does not dynamically change at any given time. Moving target defense is a new method that will allow for an active defense to stop eavesdropping (Ma et al., 2016). In a traditional use of MTD, IP addresses or ports are changed to keep attackers from listening to the network; however, little is done to change network protocols, due to the complexity. Eavesdropping is categorized into two types of attacks, session and packet. In a session attack, the entirety of the communication session is grabbed by the attacker and then analyzed based on the network protocol. In a packet attack, a series of packets from the session are grabbed and analyzed for their source IP and destination IP; this could potentially give the attacker enough for a “man in the middle” or DDoSmattack (Ma et al., 2016).

One proposed method is to use MTD with protocol-oblivious forwarding (POF), which allows the network to simply forward the packet based on the key associated with it; otherwise, it has to parse the packet first before determining what to do with it. In this setup, the clients use dynamic message packaging and dynamic routing paths to keep the attacker confused as to the source and destination IP addresses. This proposed method will block both session and packet attacks by continually keeping the attacker guessing which bits of data fit the right network protocol.

Operational Costs

Operational costs of MTD, as mentioned, can be the actual cost of the system; however, they can also affect overall system performance, network stability, and effectiveness. Determining how much a physical system may require in capital is first based on compatibility. The physical hardware requirements for the level of security needed will increase the cost of the system. The biggest challenge to an efficient MTD system is available bandwidth. To have a secure system, the total number of channels available for the system to “hide” in directly impacts the difficulty of the attacker finding it. To clarify, if a system is designed with only 10 available channels, then an attacker has an easy time scanning all channels to find the information. However, if the system has 50 possible channels, the attacker must spend much more time trying to find the information.

Capital restrictions are a dominant driving force behind the decision to change systems. The MTD system requires constant synchronization based on CPU cycles and memory systems in the network, which could detract from the processing power the company may need to service demand; this could again end up costing actual capital revenue if handled poorly. The effects on performance metrics may also lead to a loss of availability in the system. In a CPU system, it is possible to overclock the synchronization; however, it is not necessarily a best practice and can lead to system failures.

Another operational cost of MTD is the effectiveness of the system itself. Considering an MTD system with many access points and changes happening in the system, the controller has a very complex role in determining when and how requests should be handled. Deploying a large network to handle minimal security work would be wasteful and would be impossible to secure highly classified data in a small system.

The main method of determining the right cost functions of deploying an MTD is by observing possible network parameters. First, classification and value assignment of the system should be taken care of, and then experimental bandwidth consumption can be handled (Leeuwen, Stout, & Urias, 2015). The classification step determines required work factors, including operating expenses, capital expenses, performance in either network or host-based and applications, service impacts and scalability. Depending on the classification, a metric and unit will be assigned. The metric for operating expenses is operator workload, and the unit is physical man-hours. For scalability, the number of nodes and count is the metric and unit, respectively. This classification system leads to a concise requirement and possible prediction of whether a large or small network can be handled.

Observing the physical and cyber costs of deploying a new system is crucial when defending the use of new technology. There are very clear advantages to studying these metrics and optimizing where necessary to provide the strongest case as to why this technology is needed, other than for more secure data transfer.

Obfuscation of the Attack Surface

A key advantage to using software-defined networks in moving target defense is the ability to obfuscate the attack surface. Using software-defined radios, it is possible to change each network characteristic to protect it. Two of proposed network attacks to protect from are network reconnaissance and OS fingerprinting (Kampanakis, Perros, & Beyene, 2014). An SDN controller monitors the traffic coming through a network. If the traffic is malicious, the controller will attempt to quarantine that part of the network by blocking off that group of devices. The main process of this defense is to open more possible ports than are already open, causing the attacker to search more possible entries before finding the actual port. If the attacker attempts to find network configurations through an HTTP GET eavesdropping method, it is possible to change different operating system information. The httpd service will work with the SDN controller to create a dummy service version.

The network firewall is normally used to keep out attackers by preventing attacks on operating system information. However, the SDN will reassemble TCP into a spoofed version that will exist on the network; users who have access by presenting the correct key will be given the correct information from the SDN controller. In the SDN, route mutation and host randomization can also provide ideal possibilities for keeping an attacker from finding the correct path. Route mutation adds problems to the attacker in operational cost; this is since they would now be aiming randomly when sniffing packets. If they wanted to be more effective, they would need more robust machines and algorithms.

Many proposed algorithms present algorithms that make it harder for an attacker to get into the network; however, there is generally a problem for network defense when using high bandwidth applications (Li, Dai, & Zhang, 2014). The focus is to ensure that the network can morph to prevent an attack while maintaining a time-sensitive goal for transmission set by the application. Creating a real-time traffic morphing algorithm requires three pieces that work in the algorithm. The first is to create an adaptive packet generation, next is maintaining deadlines in the packet generation scheduler, and last is to minimize redundancy. Adaptive packet generation is responsible for maintaining uniqueness in the system. The deadline schedule is responsible for not allowing the first part of the algorithm from taking too long or from generating packet combinations that will cause overhead. Minimizing redundancy is a final check that the system overhead does not take too long.

Cloud Controller Characteristics

Many common open-source cloud controllers have been used for synchronization in MTD systems. One of the most prevalent is OpenFlow API, which can work as a load balancer, handling requests on a round robin basis. It works by picking from the pool and handling each request. The main method of changing specific network characteristics is either by using an encrypted pseudorandom sequence between both the transmitter and receiver continuously or simply using a lookup table to keep track of the changes on both ends; the latter is easier but the table could be vulnerable, eventually (Corbett et al., 2014).

After the network controller has been set up, the next stage is to determine how packets will be sent from transmitter to receiver. Cloud-based network systems offer a potential increase in the effectiveness of software-defined radio. This is because of the ability to run larger applications on the cloud, while the radios can perform smaller but more rapid actions (Debroy, Calyam, Nguyen, Stage, & Georgiev, 2016). Utilizing a cloud-based system can amplify vulnerability detection by covering more of the attack surface; however, it also may become a target of attack. Combining SDN and cloud adds complexity to an attacker’s attempts by splitting what can be taken away from the system. Realizing a complete network that will utilize cloud-based systems offers challenges in both total resource consumption and effective performance of the network. The operational cost of a system can significantly influence the actual utilization of a system; if the cost-to-resource usage ratio is not perfect, there is an increased potential for incurred financial losses for the service provider. In the presence of an attack, the controller will attempt to mark the attack path of IP addresses and block the attack by severing the connection; the controller will either be proactive or reactive, based on the advance of the attacker in the network.

Proposed Solution

The goal of the system is to add another layer of complexity to frequency hopping using packet fragmentation, with the intentions to thwart two very popular attack types, DoS and packet sniffing. An N×N multiple-input-multiple-output (MIMO) system utilizes the ability to transmit and receive large amounts of data at a given time. Another advantage is the ability to spread the frequency spectrum across the three pairs. Using coordinated universal time, UTC, all pairs will know precisely when to hop to another frequency or request an entirely new array of frequencies. The frequency lists are randomly generated by the transmitter and synchronized with the receiver, using a cloud-based controller. Also, the frequency of the updating the lists is randomized to add a layer of security to the operation of the cloud-based controller beyond the channel encryption.

Packet Fragmentation with Frequency Hopping

The LabVIEW program to initialize each universal software radio peripheral (USRP) device to transmit each fragment follows the logic displayed in Figure 2.

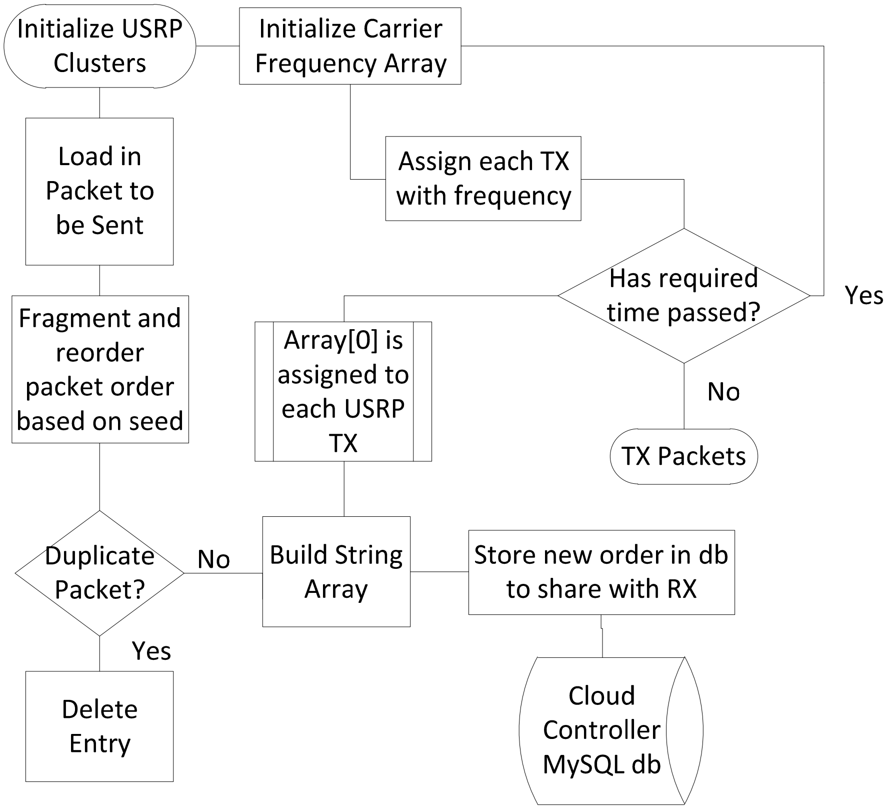
Initialize USRP Clusters
Initialize Carrier Frequency Array
Load in Packet to be Sent
Assign each TX with frequency
Has required time passed?
Yes
No
Fragment and reorder packet order based on seed
Array[0] is assigned to each USRP TX
TX Packets
Duplicate Packet?
No
Build String Array
Store new order in db to share with RX
Yes
Delete Entry
Cloud Controller MySQL db

Figure 2. Flow diagram for transmission of packet.

Each time the program loops, the packet will be split into a pseudorandom order and be given to the transmitter. Simultaneously, the transmitter is prepped to take the packet by selecting its frequencies and reporting those frequencies to the cloud-based controller's database. Each transmitter can send a varying number of total packets. In this case, the attacker would need to determine the system frequency, the total size of each packet, and the order of sending; all the while, at a designated time interval, all three of those parameters change to a new value. Synchronization between the transmitter and receiver can be achieved by using the database controller to store the current configuration set by the transmitter and synchronize it with the receiver. The cloud controller can serve as the central source, so both transmitter and receiver are given all parameters. Once ready, the USRP will continuously transmit the packet until all the frequencies are used. The program then halts, selects new frequencies, scrambles the packet in a new order, and continues.

3x3 MIMO Connection

A 3×3 MIMO system utilizes the ability to transmit and receive large amounts of data at one given time. Another advantage of this MIMO system, when used as one-to-one SISO pairs, is the ability to spread the frequency spectrum across the pairs and sacrifice bandwidth to increase data security. Using UTC, all pairs will know precisely when to hop to another frequency or request an entirely new array of frequencies. Figire 3 shows the system setup for a 3×3 USRP connection.

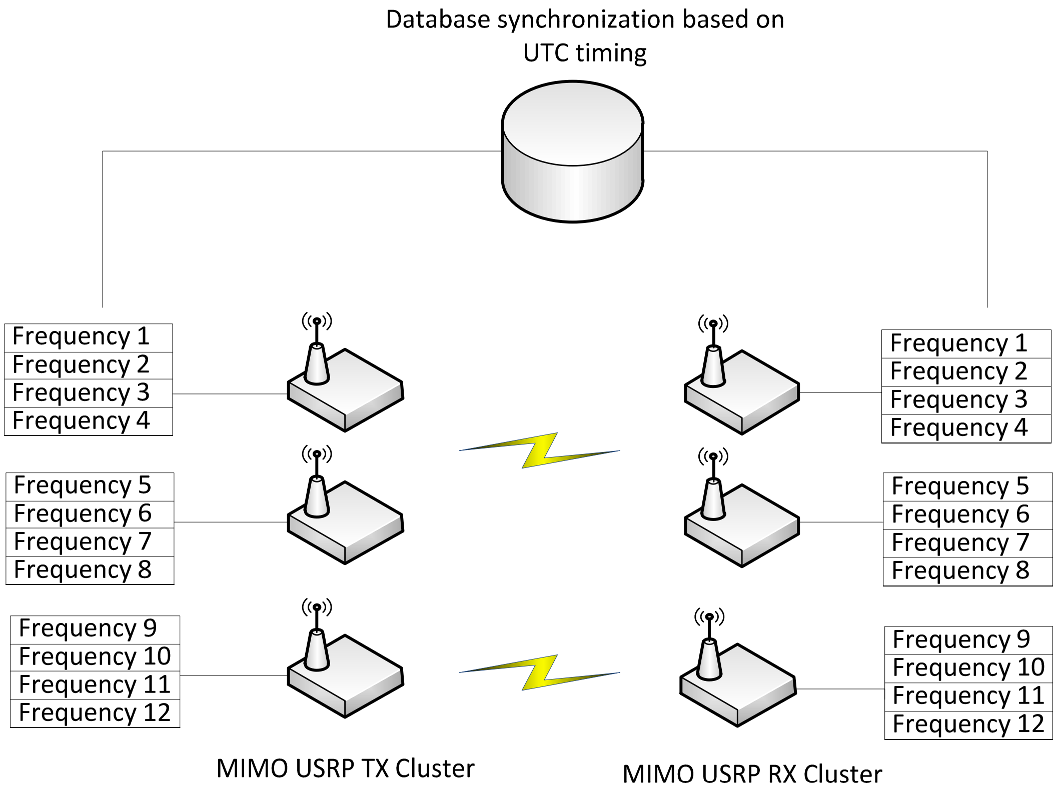

Figure 3. Structure of the 3x3 MIMO USRP network with cloud database controller.

Experimental Results

Experimental Design Specification

Figure 4 is the system setup for the Tx/Rx pair. When the experiment begins, the system receives a predefined number of allowed channels (10, 25, or 50 channels); then the frequencies are pseudo-randomly mixed, and three frequencies are assigned for transmission. A hopping interval is also predetermined (10, 30, or 60 seconds) when the system begins; at the end of the hopping interval, the next frequency is selected. Once all three frequencies are used, the system will re-randomize the frequency list and select another three frequencies to be used.

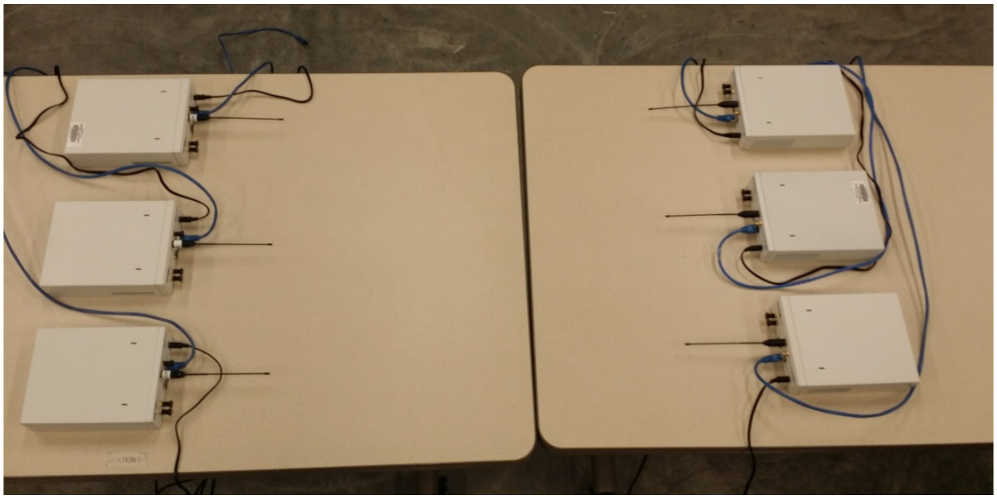

Figure 4. 3x3 MIMO USRP experimental setup.

The eavesdropper will need to know or guess the possible length of the spectrum and the packet size. The attacker will then have to scan through the entire list of frequencies from start to finish, hoping to find the correct frequency. As mentioned previously, the packet size influences the attacker's hopping speed; if it is too fast, it may miss most of the data. In a real-world application, the packet size would be considered the right amount if the attacker knew at least how much to expect; it would simply stay connected to the network until it felt as though it has collected enough data to make up the whole packet. The attacker will scan the spectrum frequencies with a hopping interval of three seconds in an incremental fashion.

Frequency Hopping WITHOUT Packet Fragmentation

In the first experiment, one transmitter was set up to broadcast a packet on varying frequencies to one receiver. In this case, the attacker will focus on finding the frequency that both are currently on and steal the packet. The attacker will sweep the network as quickly as possible to steal the packet. This experiment was tested three times, set to varying number of available channels. In each trial, the hopping interval of the Tx/Rx pair was set to either 10, 30, or 60 seconds. The attacker scanned the entire network 100 times, checking each channel for the packet, in this case, a simple "Hello World!" message. The relative frequency, *f*, of obtaining a packet is used to model the empirical probability *p* of successful eavesdropping using the following model:

$$f = \frac{n_c}{N} \tag{1}$$

Where n_c is the number of captured packets, N is the total number of packets transmitted. When the total number of packet approaches infinity, the relative frequency will converge to model the probability of eavesdropping as follows:

$$p = \lim_{N \to \infty} f \tag{2}$$

Table 1 displays the results from each combination of channels and hopping intervals.

Table 1. Empirical probability of successful eavesdropping using various system parameters.

Hopping Interval (sec)	10 Channels	25 Channels	50 Channels
60	$11.73 * 10^{-3}$	$4 * 10^{-3}$	$1.76 * 10^{-3}$
30	$7.33 * 10^{-3}$	$3.46 * 10^{-3}$	$1.68 * 10^{-3}$
10	$12 * 10^{-3}$	$2.08 * 10^{-3}$	$0.8533 * 10^{-3}$

Experimental probabilities are determined by the total successful attempts by the attacker divided by the total number of packets sent during the transmission.

Frequency Hopping WITH Packet Fragmentation

The next experiment was designed to test the ability of an eavesdropper to steal the entirety of the data being transmitted when it is broken into multiple parts and sent across three transmitters. For this trial, the network size was 25 channels, while the hopping time intervals stayed the same from the previous experiment. The results in Table 2 show that when the packet is split into multiple parts, the probability of the attacker getting the entirety of the message decreases across all three of the hopping intervals.

Table 2. Empirical probability of successful eavesdropping with/without fragmentation for 25 channel system.

Hopping Interval (sec)	With Fragmentation	Without Fragmentation
60	$2.72 * 10^{-3}$	$4 * 10^{-3}$
30	$2.187 * 10^{-3}$	$3.46 * 10^{-3}$
10	$1.76 * 10^{-3}$	$2.08 * 10^{-3}$

Conclusion

Frequency hopping provides a level of security to a network system; however, as Table I shows, it alone is not a full-proof method. Eavesdropping has been shown to be an easy and very cheap method to achieve generally. This method added another layer of confusion to the system, causing the attacker to work significantly harder to steal the information and increasing the time and resources of the attacker. The results from experimentation show that the difficulty of an eavesdropping attacker to recover the packet increases as the system covers more of the spectrum at random frequencies.

It is also essential to note that with a limited number of channels, as in Table 1, the system pair may end up hopping too often and cross the attacker more frequently. This occurred with this system when hopping between 10 channels every 10 seconds. At the same time, having a system with a very large spectrum may incur more substantial, infeasible operational costs. The proposed method can save on spectrum space by fragmenting. The results also show the effectiveness of the packet fragmentation method on bandwidth spectrum allocation. The probability of a successful attack when the system had 50 possible channels was significantly lower than when at 25 or 10; however, occupying that many channels may incur very high operational costs. Using packet fragmentation, the probability of success by using 25 channels instead of 50 yielded a difference of $0.96*10^{-3}$, $0.507*10^{-3}$, and $0.9067*10^{-3}$ for hopping intervals 60, 30, and 10, respectively. This showed the ability to be at comparable levels of security while existing on half as much total bandwidth.

References

Aydeger, A., Saputro, N., Akkaya, K., & Rahman, M. (2016). Mitigating crossfire attacks using SDN-based moving target defense. *Proceedings of the IEEE 41st Conference on Local Computer Networks*. Piscatawny, NJ: IEEE.

Cai, G., Wang, B., Luo, Y., Li, S., & Wang, X. (2016). Characterizing the running patterns of moving target defense mechanisms. *Proceedings of the 8th International Conference on Advanced Communication Technology*. Piscatawny, NJ: IEEE.

Corbett, C., Uher, J., Cook, J., & Dalton, A. (2014, March). Countering intelligent jamming with full protocol stack agility. *IEEE Security Privacy, 12*(2), 44-50.

Debroy, S., Calyam, P., Nguyen, M., Stage, A., & Georgiev, V. (2016). Frequency-minimal moving target defense using software-defined networking. *Proceedings of the International Conference on Computing, Networking and Communications*. Piscatawny, NJ: IEEE.

Green, M., MacFarland, D., Smestad, D., & Shue, C. (2015). Characterizing network-based moving target defenses. *Proceedings of the Second ACM Workshop on Moving Target Defense*. New York: ACM.

Kampanakis, P., Perros, H., & Beyene, T. (2014). SDN-based solutions for moving target defense network protection. *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. Piscatawny, NJ: IEEE.

Leeuwen, B. V., Stout, W., & Urias, V. (2015). Operational cost of deploying moving target defenses defensive work factors. *Proceedings of the Military Communications Conference*. Piscatawny, NJ: IEEE..

Li, Y., Dai, R., & Zhang, J. (2014). Morphing communications of cyber physical systems towards moving-target defense. *Proceedings of the IEEE International Conference on Communications*. Piscatawny, NJ: IEEE.

Ma, D., Wang, L., Lei, C., Xu, Z., Zhang, H., & Li, M. (2016). Thwart eavesdropping attacks on network communication based on moving target defense. *Proceedings of the 35th International Performance Computing and Communications Conference*. Piscatawny, NJ: IEEE.

Yeung, F., Cho, P., Morrell, C., Marchany, R., & Tront, J. (2016). Modeling network based moving target defense impacts through simulation in Ns-3. *Proceedings of the Military Communications Conference.* Piscatawny, NJ: IEEE.

Biographies

ISAAC J. CUSHMAN is a graduate student in the Department of Electrical and Computer Engineering at Georgia Southern University, Statesboro, GA. He received his BS degree in Electrical Engineering from Georgia Southern University in 2016, and his MASE degree in Electrical Engineering from Georgia Southern University in 2017. His research interests

include wireless communication, cyber-security and cyber-physical systems, networks and cloud computing. Isaac may be reached at ic00214@georgiasouthern.edu.

RAMI J. HADDAD is currently an associate professor in the Department of Electrical and Computer Engineering at Georgia Southern University, Statesboro, GA. He received his BS degree in Electronics and Telecommunication Engineering from the Applied Sciences University, Amman, Jordan, in 2004, his MS degree in Electrical and Computer Engineering from the University of Minnesota Duluth, Duluth, MN, in 2006, and his PhD degree from the University of Akron, Akron, OH, in 2011. He is an IEEE senior member. He is the founding director of the Optical Networks and Smart Grid Applications Laboratory, Georgia Southern University. His research focuses on various aspects of optical fiber communication/networks, broadband networks, multimedia communications, UAV ad hoc networks, cyber-physical systems, multimedia bandwidth forecasting and engineering education. Dr. Haddad may be reached at rhaddad@georgiasouthern.edu.