

FORTIFYING THE PHYSICAL LAYER OF WLANS

by

STANLEY LEE CEBULA III

B.S., May 2009, Virginia Wesleyan College

A Thesis Submitted to the Faculty of

Norfolk State University

in Partial Fulfillment of the

Requirements for the degree of

MASTER OF SCIENCE

COMPUTER SCIENCE

NORFOLK

MAY 2011



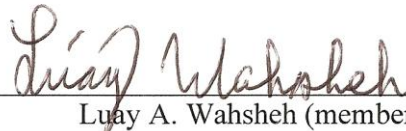
5.4.2011

Aftab Ahmad (director)



5/5/2011

Jonathan M. Graham (member)



5/4/11

Luay A. Wahsheh (member)

ABSTRACT

FORTIFYING THE PHYSICAL LAYER OF WLANS

Stanley Lee Cebula III
Norfolk State University, 2011
Director: Dr. Aftab Ahmad

The IEEE 802.11-2007 protocol, like its earlier versions, provides a robust MAC layer with the help of mandatory CCMP and comprehensive key-generation, derivation, and distribution mechanisms. The MAC layer and physical layer has not quite achieved the confidence of the networking community at the level of IPsec as of yet. We present the results of a study that looks at the IEEE 802.11-2007 MAC security in juxtaposition to IPsec. While the IEEE 802.11-2007 protocol has a strong MAC layer, the physical layer continues to be without any protection from signal privacy attacks and anonymity of attacker within the WLAN, and it has no solution in the standard. In order to combat these security holes, we present a signal strength monitoring system (SSMS) and location determination system (LDS) that increase the security of WiFi networks on the physical layer. In order to develop an accurate SSMS, an accurate channel model is needed. We discuss the Okumura-Hata Model, Log-distance Path Loss Model, and JTC Indoor Path Loss Model. However, we found that with the help of measurements spread over an extended period of time, these popular channel models did not match our environment. We developed our own empirical channel model to use for the SSMS. Finally, we discuss four location determination mechanisms for WLANs based on geolocation models (Global Positioning System, Angle of Arrival, Time-based Models and a Signal Strength based model). We also present a new mechanism that determines the location of a WLAN station within a triangle or quadrangle with probabilistically the strongest

vertices. The Signal Strength based model is identified to be the most appropriate LDS in a WLAN, because of its proven accuracy. Such an algorithm can be used to locate all users in an infrastructure type of WLAN, including an attacker. Thus, the SSMS and LDS make WiFi networks more secure.

ACKNOWLEDGEMENTS

First, I would like to thank my wife, Susana. Without your love and support, I would not have made it very far. I look forward to our very long and happy life together. Next, I would like to thank my parents. Thank you for raising me the way you did. Thank you for instilling me with a strong work ethic and determined, successful attitude. I would also like to express my gratitude towards the rest of my family. I appreciate your support and encouragement.

I would like to thank my thesis advisor, Dr. Aftab Ahmad, for his patience, resourcefulness, and expertise. Thanks to the rest of my thesis committee, Dr. Jonathan Graham and Dr. Luay Wahsheh, for your flexibility and creative ideas. I am also very thankful for the Massie grant, which funded all of my research, and IA-REDI, which provided me with a place to conduct my research.

Lastly, I would like to acknowledge Steve Park and Dave Geyer who are the authors of the code I used for random number generation.

TABLE OF CONTENTS

TITLE AND APPROVAL PAGE.....	i
ABSTRACT.....	ii
ACKNOWLEDGEMENTS.....	iv
TABLE OF CONTENTS.....	v
LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
CHAPTER 1: INTRODUCTION.....	1
1.1: Description of WiFi Security.....	1
1.1.1: RSNA Security Algorithm Framework.....	1
1.1.2: RSNA Confidentiality Protocols.....	2
1.1.3: RSNA Security Association Management.....	6
1.1.4: RSNA Keys and Key Distribution.....	7
1.2: Attacks Against WiFi.....	14
1.2.1: Session Hijacking Attacks.....	14
1.2.2: Denial-of-Service Attacks.....	15
1.2.3: Man-in-the-Middle Attacks.....	17
1.2.4: Forgery Attacks.....	18
1.2.5: Other Attacks and Security Holes.....	19
1.3: Problem Statement and System Architecture.....	21
1.3.1: Environment Design.....	22
1.3.2: Proposed Solution.....	22
1.3.3: System Architecture.....	23
CHAPTER 2: CHANNEL MODELING.....	25
2.1: Factors in Modeling Wireless Channels.....	25
2.2: Existing Path Loss Models.....	26
2.2.1: Okumura-Hata Model.....	26
2.2.2: Log-distance Path Loss Model.....	27
2.2.3: JTC Indoor Path Loss Model.....	28
2.3: Development of Custom Channel Model.....	30
2.4: Selection of Channel Model.....	33
CHAPTER 3: SIGNAL STRENGTH MONITORING SYSTEM.....	35
3.1: Program Overview.....	35
3.2: Static vs. Simulated Real-Time vs. Real-time.....	35
3.3: Program Components.....	36
3.3.1: Struct.....	36

3.3.2: Draw Functions.....	36
3.3.3: Finding the Distance.....	37
3.3.4: Calculate Signal Strength.....	37
3.3.5: Signal Strength to Color Conversion.....	37
3.3.6: Color Each Pixel.....	37
3.3.7: Output Information to File.....	38
3.4: Output Examples.....	38
3.5: Results.....	40
CHAPTER 4: LOCATION DETERMINATION SYSTEM.....	41
4.1: Geolocation Models.....	41
4.1.1: GPS.....	41
4.1.2: AOA.....	42
4.1.3: Time-based Models.....	43
4.1.4: Ahmad's Algorithm of Closest Vertices.....	44
4.1.5: SS based.....	46
4.2: Selection of Geolocation Model for LDS.....	48
4.3: Results.....	50
CHAPTER 5: CONCLUSION AND FUTURE WORK.....	51
5.1: Conclusion.....	51
5.2: Future Work.....	52
REFERENCES.....	53
APPENDIX A – PROGRAM CODE.....	56

LIST OF TABLES

Table 1: Log-distance Path Loss Exponent.....	28
Table 2: JTC Indoor Path Loss Model Variables.....	29
Table 3: Distance of Locations from Access Point.....	31
Table 4: Average Signal Strength Compared to Distance.....	33
Table 5: Signal Strength to Color Conversion.....	38

LIST OF FIGURES

Figure 1: MAC Protocol Data Unit Using CCMP.....	3
Figure 2: CCMP Encapsulation Process.....	4
Figure 3: CCMP Decapsulation Process.....	5
Figure 4: Pairwise Master Key Structure.....	8
Figure 5: Supplicant State Machine.....	11
Figure 6: Authenticator State Machine.....	12
Figure 7: WiFi Security by Internet Layer.....	21
Figure 8: System Architecture with Two Clients Connected.....	23
Figure 9: Experiment Environment Overview.....	31
Figure 10: Cumulative Distribution Function of Collected Data.....	32
Figure 11: Signal Strength Compared to Distance.....	32
Figure 12: Static Custom Channel Model (no randomness) Output.....	38
Figure 13: JTC Indoor Path Loss Model (with randomness) Output.....	39
Figure 14: Static Custom Channel Model (with randomness) Output.....	39
Figure 15: GPS Satellite Constellation.....	42
Figure 16: AOA Measurements and Calculations.....	42
Figure 17: Sensor Grid with Quadrants.....	46
Figure 18: SS Location Determination Process.....	47

1. INTRODUCTION

1.1 Description of WiFi Security

In order to understand the security that the IEEE 802.11 standard provides for WLANs, we are going to look at an extended service set model only. Also, all of the devices used in the network are assumed to be technologically up-to-date. The extended service set network structure represents the basic infrastructure network where there is one or more access points and many clients connected to it. The security provided by the IEEE 802.11 standard defines two types of classes of security algorithms: RSNA (robust security network association) algorithms and pre-RSNA algorithms. Since pre-RSNA security algorithms (like WEP or wired equivalent privacy) have been deprecated, they will not apply to this study. Therefore, only RSNA algorithms will be considered.

The rest of this section will summarize and discuss the security provided by the IEEE standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. The areas that will be discussed include: RSNA security algorithm framework, RSNA confidentiality protocols, RSNA security association management, and RSNA keys and key distribution.

1.1.1 RSNA Security Algorithm Framework

The security of RSNAs can be affected by the environment in which it is used and the correctness of assumptions and constraints. RSNAs can be established in an extended service set that is based on authentication and key management and an extended service set that is based on a preshared key. RSNAs also have a set of assumptions and constraints that increase security: each station can generate cryptographic-quality random numbers (randomness is required in cryptography), mutual authentication is used

(prevents man-in-the-middle attacks), the mutual authentication method must be strong (just strong enough to make impersonation attacks computationally infeasible), the access point and authentication server has a secure channel between them (preventing exposure of cryptographic keys), the authentication server keeps the symmetric key used by the access point and station a secret (ensuring the authentication server is never compromised), a station keeps the common symmetric key used with its peer a secret (preventing attackers from gaining access to a means of breaking the cipher used), the station's supplicant and authenticator generate a new pairwise transient key for each new session (meaning no keys are reused), and ARP (address resolution protocol) or ICMP (internet control message protocol) is used to ensure the destination station is the correct party (making sure nothing is sent by accident to a different party) [1]. All of these assumptions and constraints are important to making RSNA algorithms as secure as possible. The environment that RSNAs are used in and the use of constraints and assumptions increase the security of 802.11 networks.

1.1.2 RSNA Confidentiality Protocols

The IEEE standard defines two data confidentiality protocols for 802.11 networks: TKIP (Temporal Key Integrity Protocol) and CCMP (counter mode with cipher-block chaining message authentication code protocol). CCMP must be used in all devices that claim they are RSNA compliant. TKIP is only used when communicating with devices that are not able to communicate with CCMP (older devices based on pre-RSNA security). We assume CCMP will be used.

The purpose of CCMP is to provide confidentiality, integrity, and authentication on RSNA devices. CCMP is based on the counter mode with cipher-block chaining

message authentication code of the AES (advanced encryption standard) encryption algorithm. It combines a counter mode for data confidentiality and cipher-block chaining message authentication code for authentication and integrity. This protects the integrity of the MAC protocol data unit data field and some parts of the MAC protocol data unit header. Pictured in Figure 1 is a picture of the MAC protocol data unit when using CCMP. The highlighted sections of the MAC protocol data unit represent the additions CCMP makes to the original data unit. CCMP uses a message integrity code and encryption in order to protect the data field. The message integrity code is calculated over the source address, destination address, priority parameter of the message, and the unencrypted data in the message. There is an encapsulation process which encrypts the MAC protocol data unit before it is sent and a decapsulation process which decrypts the MAC protocol data unit upon reception. The CCMP header includes the use of a 6 octet pseudonoise code sequence, reserved octet, and key ID octet. The reserved octets and bits are set to zero and ignored. All of the other fields will be used in the encapsulation and decapsulation processes. The CCMP encapsulation process is used to secure information during transmission. Figure 2 represents the CCMP encapsulation process.

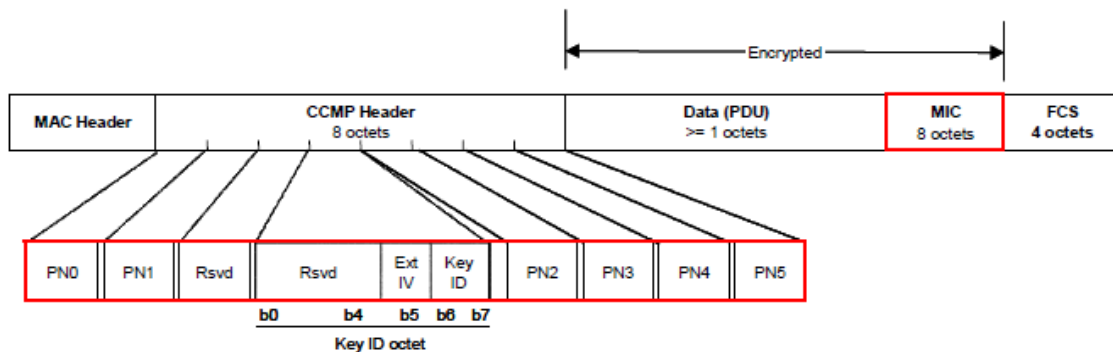


Figure 1 - MAC Protocol Data Unit using CCMP [1]

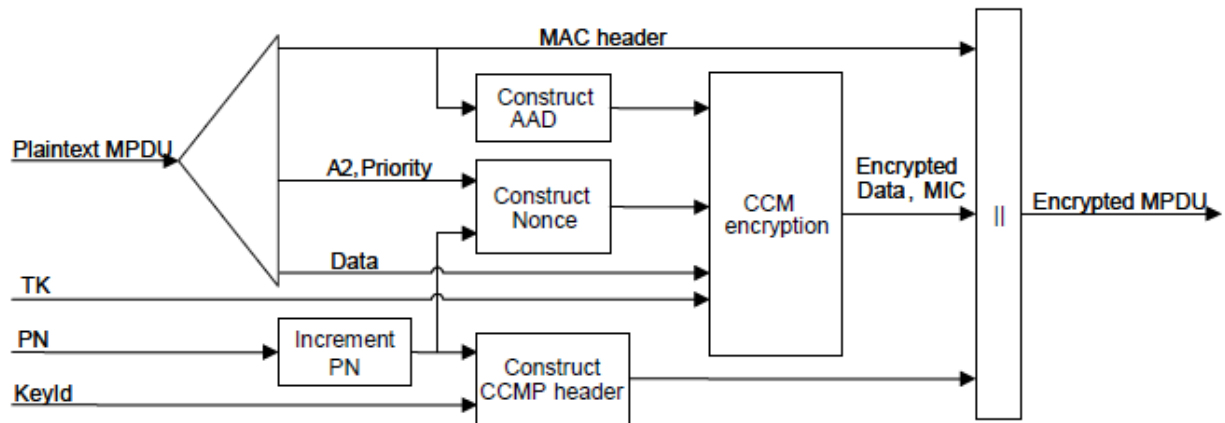


Figure 2 - CCMP encapsulation process [1]

CCMP encrypts the plaintext MAC protocol data unit and then encapsulates the cipher text. First, the pseudo-noise code sequence is incremented so that it never repeats for the same temporal key. Second, the additional authentication data is constructed using information in the MAC header. This information will be used later in order to verify authentication. Next, the CCM nonce block is constructed from the pseudo-noise code sequence, A2 (MAC protocol data unit address 2), and priority fields. Then, the new pseudo-noise code sequence and key identifier are placed into the CCMP header. Next, CCM originator processing is completed which includes producing the cipher text and message integrity code using the temporal key, additional authentication data, nonce value, and MAC protocol data unit data. Finally, the encrypted MAC protocol data unit is configured using the original MAC protocol data unit header, CCMP header, encrypted data, and message integrity code. This MAC protocol data unit is now safe for transfer.

In order to decrypt the encrypted MAC protocol data unit, CCMP decapsulation needs to take place. The CCMP decapsulation process is illustrated in Figure 3.

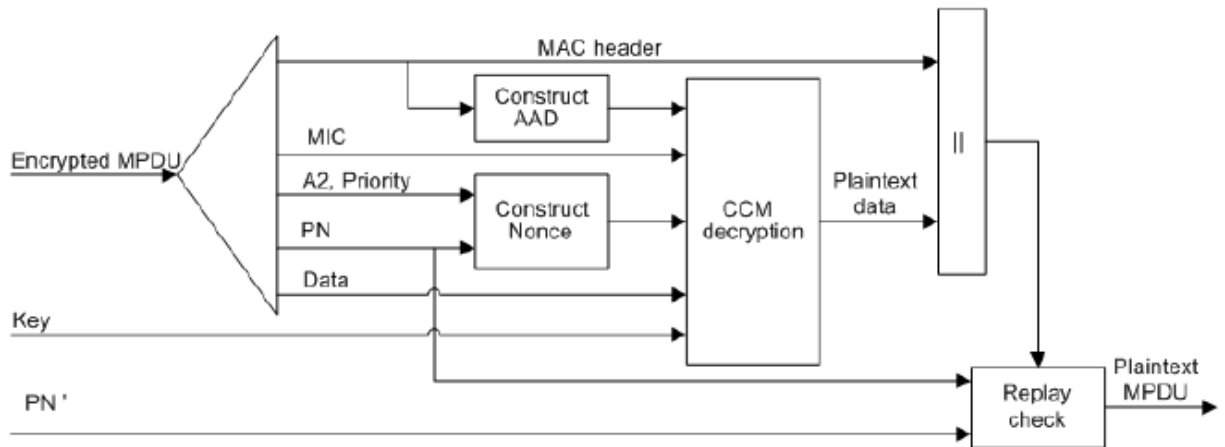


Figure 3 - CCMP decapsulation process [1]

CCMP decrypts the encrypted MAC protocol data unit and decapsulates the plaintext.

The decryption keys are only known by the supplicant and authenticator. First, the encrypted MAC protocol data unit is parsed to retrieve the additional authentication data and nonce values. Second, the additional authentication data is constructed from the MAC protocol data unit header of the encrypted MAC protocol data unit. Next, the nonce value is formed using the A2, pseudo-noise code sequence, and priority fields. Then, the message integrity code is retrieved in order to conduct CCM integrity checking. Next, the CCM recipient processing recovers the MAC protocol data unit plaintext data using the temporal key, additional authentication data, nonce value, message integrity code, and MAC protocol data unit cipher text. Also, the integrity of the additional authentication data and MAC protocol data unit plaintext data is checked. Finally, the MAC protocol data unit header and MAC protocol data unit plaintext are combined to form a plaintext MAC protocol data unit. Now, information can be successfully retrieved.

1.1.3 RSNA Security Association Management

Security association means secure operations. A security association is a set of policies and keys used to protect information. Each party in the security association stores this information. There are two types of security associations supported by an RSN station that could be used in a classified environment¹: pairwise master key security association and pairwise transient key security association.

A pairwise master key security association occurs upon a successful IEEE 802.1X exchange, preshared pairwise master key information, or when the pairwise master key is cached [1]. This security association is bidirectional, which means both parties can send and receive information. This security association is also used to create a pairwise transient key security association.

A pairwise transient key security association results from a successful four-way handshake. This security association is bidirectional. There is only one pairwise transient key security association binding a pair of supplicant and authenticator MAC addresses.

In order to establish an RSNA with other stations, a station must advertise its capabilities and specify all the authentication and cipher suites enabled by their policies. The policy selection, authentication, and key management processes are all important when considering the security of the communication between two stations.

The station attempting to connect to another station performs RSNA policy selection. In order for communication to be successful, the connecting station must use these policies. Stations that wish to authenticate will only do so to other stations that they choose to connect to. This prevents unwanted and distrusted connections. Before

¹ Throughout this thesis report, we assume that a WLAN in a classified environment is considered.

authentication is completed, all communications between the authenticator and supplicant are completed through the IEEE 802.1X uncontrolled port. After authentication is successful, authenticators and supplicants can communicate using the IEEE 802.1X controlled port. After authentication has been completed, the station and authentication server will share a secret key called a pairwise master key. After the pairwise master key has been shared, a key confirmation handshake is initiated by the four-way handshake in order to: confirm the existence of the pairwise master key for the peer, ensure the security association keys are fresh, synchronize the installation of temporal keys into the MAC, transfer the group temporal key from the authenticator to the supplicant, and confirm the selection of cipher suites that will be used [1]. Policy selection, authentication, and key management are all important concepts that contribute to the overall security in a security association.

1.1.4 RSNA Keys and Key Distribution

In an RSNA, there are three types of key hierarchies: pairwise key hierarchy, group temporal key hierarchy, and PeerKey key hierarchy. All key hierarchies use keys based on a pseudo-random function with an output of 128, 192, 256, 384, and 512 bits. We will only discuss the pairwise key hierarchy in this thesis.

The pairwise key hierarchy is used to protect communications from one device to another. It is based on a pseudo-random function with an output of 384 bits or 512 bits, which derives keys for each session based on a pairwise master key. Figure 4 depicts the structure of the pairwise master key. The pairwise master key (originally 256 bits) is then expanded to create a pairwise transient key (can be many different lengths). The pairwise transient key is then partitioned into three smaller keys: EAPOL (Extensible

Authentication Protocol over LANs)-key confirmation key (bits 0-128 of pairwise transient key), EAPOL-key key encryption key (bit 128 of pairwise transient key), and temporal key (bits 256-128 of pairwise transient key). Also, the temporal key will be used as the CCMP key for MAC service data units between the two stations who are communicating. These three keys are used by the MAC to protect one to one communication between the authenticator and supplicant. Each pairwise transient key is only used between two specific devices.

In order for two devices to be able to communicate securely, a four-way handshake is used. A four-way handshake is initiated at the end of a successful IEEE 802.1X authentication, after the preshared key authentication is negotiated, when a cached pairwise master key security association is used, or after a station requests a new key [1]. There are four messages in a four-way handshake, each with its own purpose.

Message one is sent from the authenticator to the supplicant. Message one sends an EAPOL-Key frame with the ANonce (nonce value the authenticator uses). The supplicant checks the key replay counter field to make sure it is greater than the current

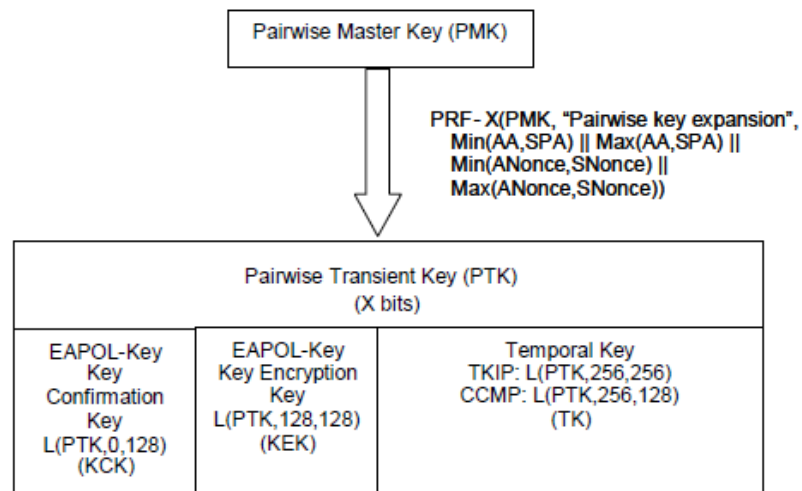


Figure 4 - Pairwise Master Key structure [1]

local value used with the pairwise master key security association. The supplicant then generates a new SNonce (nonce value the supplicant will use). After the SNonce has been generated, the pairwise transient key can be created using the ANonce and the SNonce. Message two is then constructed.

Message two is sent from the supplicant to the authenticator. The supplicant sends an EAPOL-Key frame with the SNonce, robust security network information element (from the association request frame used earlier to connect to the authenticator), and a message integrity code. The authenticator checks the key replay counter field to the counter in message one. If the key replay counter is not the same, the message is silently discarded. Otherwise, the authenticator derives the pairwise transient key using the ANonce and the SNonce. The authenticator then calculates the message integrity code. If the calculated message integrity code does not match the message integrity code listed in the EAPOL-Key frame, message two is silently discarded. If the message integrity code is valid, the authenticator checks to see if the robust security network information element is the same element used in the association request message earlier. If these elements are not exactly the same, the authenticator terminates the robust security network association between the authenticator and the supplicant. However, if the robust security network information elements match, message three is created.

Message three is sent from the authenticator to the supplicant. The authenticator sends an EAPOL-Key frame with the ANonce, robust security network information element, message integrity code, whether or not to install the temporal keys, and the encapsulated group temporal key. First, the supplicant checks the key replay counter field to make sure it matches the same field in message one. The supplicant also checks

the ANonce value in message three against message one. If the key replay counter field or ANonce value does not match, message three is silently discarded. Otherwise, the supplicant checks the robust security information element in this message against the robust security information element in the beacon or probe response message used earlier in communication. If these elements do not match, the robust security network association between the supplicant and the authenticator will be terminated. However, if these elements do match, the supplicant will calculate and check the message integrity code field. If the values match, the supplicant will update the key relay counter and construct message four. Message four is then sent to the authenticator. Lastly, the supplicant will configure its medium access control to send and receive class three unicast MAC protocol data units protected by the pairwise transient key [1]. The group temporal key is also configured.

Message four is sent from the supplicant to the authenticator. Message four is an acknowledgement message letting the authenticator know the supplicant installed the temporal keys specified in message three. The authenticator then checks the key replay counter to make sure it is the same. If it is not the same, message four is silently discarded. Otherwise, the authenticator will calculate and check the message integrity code to make sure the calculation matches what is specified in message four. If these values do not match, the authenticator silently discards message four. Otherwise, the authenticator configures its MAC to send and receive class three unicast MAC protocol data units protected by the pairwise transient key [1]. The group temporal key is also configured. Lastly, the authenticator updates the key replay counter field in case a rekey is necessary later on.

Using the four-way handshake to generate keys greatly increases the security of communication between an authenticator and a supplicant. Communication is safe if all of these steps are completed successfully.

In order to better understand the concept of the supplicant and authenticator, state machines are provided. The state machine for the supplicant has four states: authentication, disconnected, initialize, and stakeystart. The supplicant state machine is pictured in Figure 5. The supplicant will enter the authentication state when it sends an authentication request. The supplicant will enter the disconnected state when authentication fails or the device experiences a lifetime timeout. The supplicant will enter the initialize state when it receives deauthentication or deauthentication messages. Also, sometimes the supplicant will automatically enter the initialize state after being disconnected. The supplicant will enter the stakeystart state when it receives an EAPOL frame. The state machine for the authenticator has twelve states: authentication, authentication2, disconnect, disconnected, initialize, initpmk, initpsk, ptkcalcnegotiating,

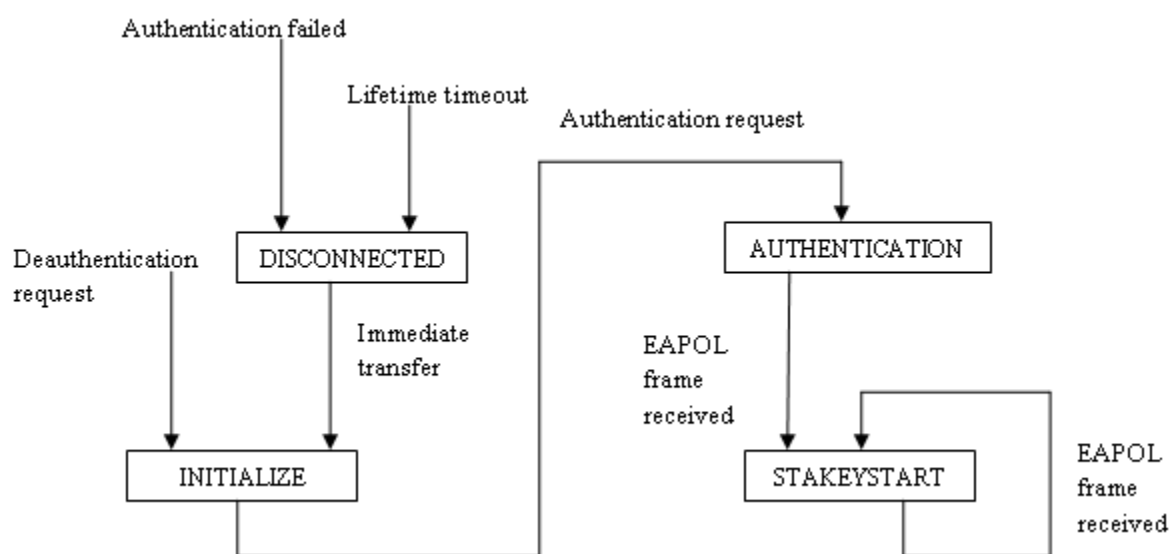


Figure 5 – Supplicant State Machine

ptkcalcnegotiating2, ptkinitnegotiating, ptkinitdone, and ptkstart [1]. The authenticator state machine is pictured in Figure 6. The authenticator will enter authentication when the authentication request is sent. The authenticator will enter authentication2 automatically from the authentication state, or if it sends a reauthentication request. The authenticator will enter the disconnect state when an EAPOL-frame fails the message integrity code check, there is a lifetime timeout, a pairwise master key is not provided in the initpmk state, or there is no response to the four-way handshake in the ptkstart state. The

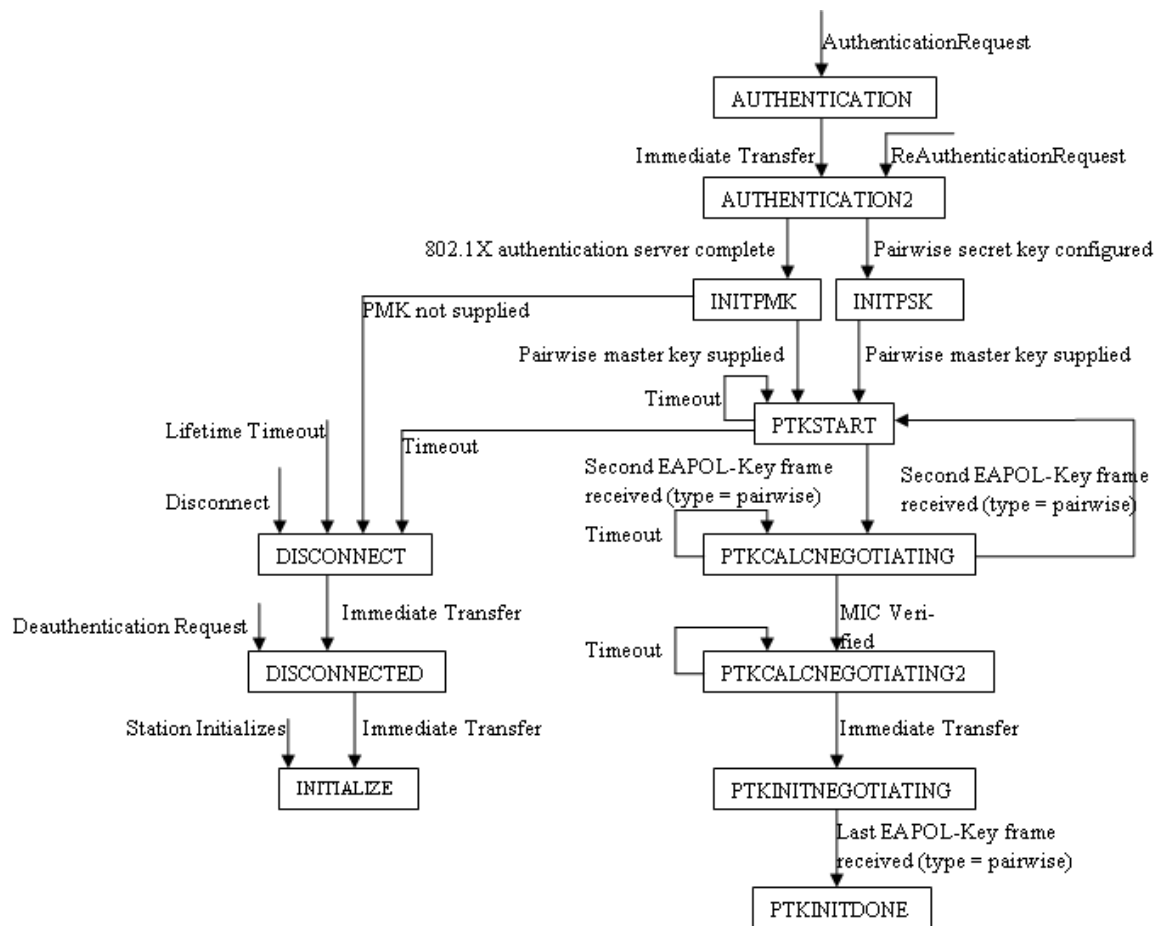


Figure 6 - Authenticator State Machine

authenticator will enter the disconnected state when disassociation or disauthentication messages are received. The authenticator will enter the initialize stage when a new station initializes and automatically from the disconnected state to try to reinitialize. The authenticator will enter the initpmk state when the IEEE 802.1X backend authentication server completes successfully. The authenticator enters the initpsk state when a pairwise secret key is configured. The authenticator will enter the ptkcalcnegotiating state when the second EAPOL-Key frame for the four-way handshake is received where the type is pairwise. The authenticator enters the ptkcalcnegotiating2 state when the message integrity code from the ptkcalcnegotiating state is verified. The authenticator will enter the ptkinitnegotiating state automatically after the ptkcalcnegotiating2 state in order to add the encrypted group temporal key at the end of message three in the four-way handshake. The authenticator will enter the ptkinitdone state when the last EAPOL-Key frame is received where the type is pairwise. Lastly, the authenticator will enter the ptkstart state from the initpmk or initpsk states in order to start the four-way handshake or if there is no response to the four-way handshake.

The last concept in keys and key distribution that needs to be discussed is nonce generation. As mentioned before, the authenticator generates the ANonce and the supplicant generates the SNonce. These variables are generated using pseudo-random functions that are 256 bits in size. Each nonce is produced with a random number, init counter, and local MAC address concatenated with the time. This ensures that each nonce value will be different on each machine. This also guarantees that each nonce value on each machine will be different, because they will be created at different times. Nonce values add a layer of security each time they are used.

1.2 Attacks Against WiFi

There are many different types of attacks and techniques that can be used to obtain information illegally from a WiFi network. This section will cover specific attacks against WiFi and some security holes in WiFi's protocol stack. Also, the prevention of these attacks will be discussed. Specifically, session hijacking attacks, denial-of-service attacks, man-in-the-middle attacks, forgery attacks, and other simple attacks will be covered that the IEEE 802.11 Standard protects against.

1.2.1 Session Hijacking Attacks

Session hijacking is gaining unauthorized access to information or services in a computer system. The most common form of session hijacking in a classified environment occurs when an attacker is trying to gain access to a network or information. In both cases, the attacker tries to gain control of a robust security network association. In order for this attack to work successfully, the attacker has to time its attacks carefully. As described in [2], "the supplicant and the authenticator engage in the authentication process, which results in the supplicant being authenticated." An attacker then sends a disassociate message to the supplicant pretending to be the authenticator (using the MAC address of the authenticator). If the supplicant trusts the message, the supplicant will disconnect from the authenticator. However, as stated in [2], "since this disassociate message was sent by the attacker, the real access point does not know about it." This attack must occur after message three is sent from the supplicant to the authenticator and before message four is sent from the authenticator to the supplicant. Therefore, the attacker can then use the supplicant's MAC address to connect to the authenticator, successfully hijacking the original session.

There are two security features built-in CCMP that prevent this attack from occurring successfully. CCMP encapsulation and encryption counter a session hijacking attempt [1, 3]. As explained in section 1.1, CCMP encrypts the plaintext MAC protocol data unit and then encapsulates the cipher text. In order for an attacker to be able to read anything transmitted in the CCMP MAC protocol data unit, the attacker would have to successfully decapsulate and decrypt it. This is not possible, because the keys needed to decapsulate and decrypt the MAC protocol data unit are only known to the authenticator and supplicant (specified in 1.1). Due to the encryption and encapsulation, the attacker does not have access to the ANonce or robust security network information element. If the key replay counter field, ANonce value, or robust security network information element is not exactly what is expected, the message will be discarded silently. Discarding messages silently prevents the attacker from realizing the attack attempt has failed. This would prevent the attacker from being able to send messages to the authenticator posing as the supplicant.

1.2.2 Denial-of-Service Attacks

A denial-of-service attack attempts to make a computer resource unavailable to other users. The most common denial-of-service attack that would occur in a classified environment is an attempt to connect over and over to the authentication server until it crashes due to network volume. This would cause all of the devices that were previously connected to the authentication server to disconnect as well. It is also reasonable to assume a denial-of-service attack could occur against a specific supplicant in order to remove them from the network. As stated in [4], “the vulnerability [of a denial-of-service attack] results from the lack of any authentication in message 1.” Denial-of-service

attacks against a specific supplicant are rare, but they can occur. In order for a denial-of-service attack to occur against a supplicant, message one would be sent over and over again.

There is a constraint in the structure of robust security network associations that protect authenticator from denial-of-service attacks. A denial-of-service attack against the authenticator will fail, because the authenticator can only have one active handshake in progress for each supplicant it is connected to. As explained in [5], the authenticator can “discard an unexpected response and retry the previous message or terminate the handshake if the expected response is not received during a given time interval and certain number of retries.” Since each authenticator can only have one active handshake per supplicant, it is impossible for a denial-of-service attack to take place.

For denial-of-service attacks against supplicants, nonce values can be reused for more protection. The 802.11i² amendment adopted a modification that allows supplicants to re-use nonce values. As asserted in [4], “re-using the nonce until one four-way handshake completes allows the supplicant to avoid storing state, which prevents memory exhaustion.” Even though reusing nonce values goes against the purpose of using nonce values (to create a new value every time), it can be successful in preventing denial-of-service attacks against the supplicant. In other words, a supplicant will not create and store a new nonce value until the original four-way handshake that created the nonce value has been completed. This will prevent hundreds and thousands of nonce values from being stored in the supplicant, which will prevent the denial-of-service attack. However, the system administrators will need to specify this option manually.

² Now a part of IEEE 802.11-2007 edition.

1.2.3 Man-in-the-Middle (MITM) Attacks

A MITM attack is generally a form of eavesdropping that takes place when an attacker fools the authenticator and supplicant into making independent connections with them acting as a midpoint. In a classified environment, eavesdropping can be very detrimental to an organization. As mentioned in [3], the attacker “fools users and other access points forcing them to send data through the unauthorized device.” However, the authenticator and supplicant think they are actually only talking to each other. The attacker can then control the entire conversation. In order for a successful MITM attack to occur, the attacker must manipulate address resolution protocol (ARP). As explained in [6], “the attacker sends the victim (supplicant) ARP replies that wrongly associate the internet protocol (IP) address of the victim’s (supplicant’s) default gateway (authenticator) with the attacker’s MAC address.” This leads the supplicant to believe they are sending packets to the authenticator. In reality, the supplicant is actually sending packets destined to the authenticator to the attacker. Furthermore, “the attacker also sends the gateway (authenticator) ARP replies that wrongly associate the victim’s (supplicant’s) IP address with the attacker’s MAC address [6].” This action will cause the authenticator to send messages intended to the supplicant through the attacker. Through successful manipulation of ARP, the attacker has setup a MITM attack.

Using random nonce values and a message integrity code in the four-way handshake prevents MITM attacks. Specifically in [1], “with unpredictable nonces, a MITM attack that uses the supplicant to precompute messages to attack the authenticator cannot progress beyond message 2, and a similar attack against the supplicant cannot progress beyond message 3.” If an attacker modifies the ANonce value or the address, it

will show in the message integrity code. This will cause the supplicant to drop every packet coming from the attacker trying to pose as the authenticator. Also, the same is true in reverse. This forces the attacker to try to pre-compute the nonce values used by the authenticator and supplicant. The use of random nonce values for the ANonce and SNonce will prevent the attacker from guessing or precomputing correctly. [6] concludes, “such verification thwarts MITM attacks.”

1.2.4 Forgery Attacks

A forgery attack consists of an attacker using false information to gain access to services or information on a computer system. The most common form of forgery in a classified environment consists of an attacker stealing a supplicant’s information (normally on the classified network), and using it to connect to an authenticator to steal information. The authenticator will think the attacker is actually the supplicant. If the attacker commits any crimes or suspicious behavior, it will be logged as coming from the supplicant as opposed to the attacker. Forgery attacks are usually used in other types of attacks. Specifically, an attacker would try to steal a supplicant’s MAC address or IP address in order to use it to connect to the authenticator. This is possible, because “IEEE 802.11i does not provide authentication of control messages for establishing association between station and access point [7].” An attacker could intercept a control message in order to determine the MAC address or IP address of the supplicant and authenticator.

Forgery attacks are prevented due to the structure of the pairwise transient key and the use of nonce values used in the encapsulation process. As stated in [1], “pairwise key support with CCMP allows a receiving station to detect MAC address spoofing and data forgery.” Even though control messages can be forged, “illegitimate control

messages prevent progress of the next stage because IEEE 802.11i carries out mutual authentication [7].” [1] continues to explain, “message 1 of the four-way handshake can be forged, however the forgery attempt will be detected in the failure of the four-way handshake.” As explained in 3.4, the nonce values that are used in the four-way handshake are created using a random number, init counter, and local MAC address concatenated with the time. Therefore, “if an attacker creates a MAC protocol data unit with a spoofed transmitting address, then the decapsulation procedure at the receiver will generate an error [1].” Since there will be an error in the decapsulation process, the attacker will never be able to successfully communicate with the authenticator.

1.2.5 Other Attacks and Security Holes

There are many attacks that are prevented through the use of a message integrity code throughout the four-way handshake process. The types of attacks that are thwarted include: bit-flipping attacks, data truncation, concatenation, and splicing attacks, fragmentation attacks, iterative guessing against the key, redirection by modifying the MAC protocol data unit destination address or receiver address field, and impersonation attacks by modifying the MAC protocol data unit source address or transmitter address field [1]. Bit-flipping attacks modify the cipher text in order to look for a pattern that arises in the plaintext. Data truncation attacks cut data off the end in order to cause errors for the receiver of the data. Data concatenation attacks put data together in order to confuse the receiver of the data. Data splicing attacks join different parts of data in order to cause errors in the decryption process. Fragmentation attacks break up a single IP datagram into smaller individual datagrams, which can cause overlapping, buffer overflow, and overwriting. Iterative guessing against the key consists of guessing what

the value of the keys could be. Redirection attacks involve editing the destination address in the MAC protocol data unit to send packets other places besides the original destination. Impersonation attacks consist of editing the source address to make the receiver think the packet came from somewhere else.

IP addresses, a message priority, and the unencrypted data are used to calculate the message integrity code. Any change to either of these between the time the sender sends the packet and the receiver receives the packet results in a different message integrity code. Bit-flipping attacks, data truncation attacks, data concatenation attacks, and data splicing attacks all modify data in a MAC protocol data unit. Fragmentation attacks alter the IP addresses used in the MAC protocol data unit. Iterative guessing against the key modifies some keys in the MAC protocol data unit. Modification of keys will result in an error when the attacker tries to compute the message integrity code. Redirection attacks modify the destination address, and impersonation attacks modify the source address. Therefore, by including the message integrity code in the four-way handshake, all of these attacks are prevented.

While WiFi protects against many attacks at the MAC layer, it does not have sufficient security at the physical layer. As depicted in Figure 7, there are specific security protocols and standards at every level of the Internet Protocol stack except for the physical layer. At the application layer, certificates and secure protocols exist to verify identities. Transport Layer Security (TLS) and Secure Sockets Layer (SSL) exist at the transport layer to provide cryptographic security. Internet Protocol Security (IPsec) operates on the network layer to guarantee confidentiality, integrity, and availability. The IEEE 802.11 standard discussed earlier performs on the link layer. However, at this point

Internet	Wi-Fi Security
Application	Certificates, HTTPS, FTPS, etc.
Transport	TLS/SSL
Network	IPsec
Link	IEEE 802.11 Section 8
Physical	

Figure 7 – WiFi Security by Internet Layer

there is no standard or accepted protocol that secures the physical layer of the Internet.

1.3 Problem Statement and System Architecture

Two specific problems at the physical layer of WiFi are addressed in this thesis.

First, attacks are anonymous at the physical layer, because there are no physical connections. An attack cannot be physically traced back to its source. This allows attackers to remain anonymous during attacks. An attacker can attack a network anywhere a WiFi signal is detected. Furthermore, attackers can remain mobile so they cannot be traced. The system administrator will be able to figure out the IP address of the attacker, but the exact physical location of the attacker will not be found. Second, the WiFi signal is uncontrolled. The signal will keep propagating until it is too weak to be picked up. If a network's signal is strong or the building housing the network is small, the signal will not stop at the walls of the building. This allows attackers to be physically outside of a building (and still in the network) they are attacking. These two problems lead to WiFi being very insecure at the physical layer.

1.3.1 Environment Design

The area that we used for testing, developing, and conducting our research is a portion of the Information Assurance Research, Education and Development Institute (IA-REDI) located on the sixth floor of the Marie V. McDemmond Center for Applied Research (MCAR) at Norfolk State University. The area we used is a computer lab (approximately twenty feet by sixty-five feet) next to an office, three conference rooms, and one long hallway. We used commercially available hardware and software. The access point we used was an Actiontec GT704WG router on default settings. The wireless card we used to connect to the access point was an Intel® PRO/Wireless 3945ABG Network Connection built-in a Dell Latitude D830 laptop on default settings. The environment is not classified, but this research is done with classified office environments under consideration.

1.3.2 Proposed Solution

In order to solve signal spilling and the anonymity of attackers at the physical layer, a tool is designed that combines a signal strength monitoring system and a location determination system. The signal strength monitoring system allows system administrators to monitor the WiFi signal presence at a physical level in order to control signal spilling. If signal spilling is occurring, system administrators will know to reduce the signal power of their network. The location determination system plots the locations of each user connected wirelessly to the network in order to prevent attacks from being anonymous. If an attack were to occur, system administrators would just need to identify the IP address on their signal map. These two systems could substantially improve WiFi's security at the physical layer.

1.3.3 System Architecture

Our system combines a signal strength monitoring system, location determination system, a grid of WiFi sensors, access point, and administrator's terminal. The signal strength monitoring system and location determination system is written in C++ (with OpenGL and GLUT) and is located on the system administrator's machine. The access point is reachable by the administrator's machine and wirelessly connected to the grid of WiFi sensors. The grid of WiFi sensors is dispersed throughout the coverage area of the WLAN. Figure 8 depicts the system architecture. The figure consists of a workstation, an access point (triangle with "AP"), a WiFi sensor grid (circles with "x"), and two

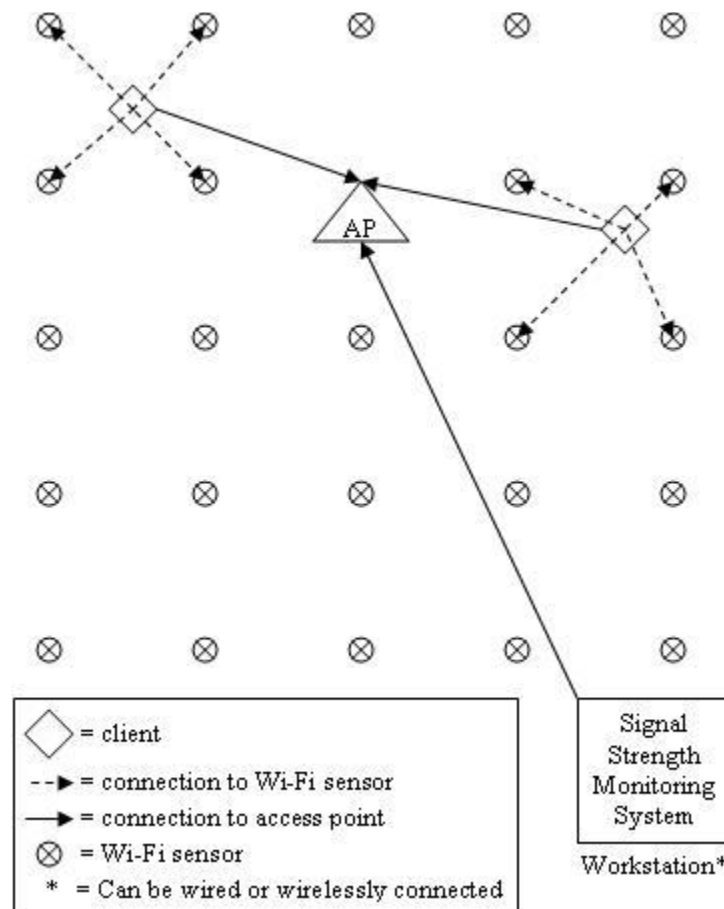


Figure 8 - System Architecture with Two Clients Connected

clients (diamonds). The dashed arrows represent connections from the client to the closet WiFi sensors. These connections will be used in the location determination system to pinpoint the client's location. The solid arrows represent connections from the client to the access point.

2. CHANNEL MODELING

2.1 Factors in Modeling Wireless Channels

In any type of WiFi signal transmission, the output signal from the transmitting station or access point will differ from the signal that is received. There are many factors that affect the signal while it is in transit. These factors include attenuation, free space loss, fading, reflection, diffraction, scattering, refraction, and noise. Attenuation occurs when the strength of a signal falls off with distance [8]. Basically, the further the signal travels, the weaker the signal will get. This can be represented logarithmically [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. The rest of this section has made heavy use of [8]. Free space loss is a form of attenuation that means the signal disperses with distance. In other words, the further the signal travels, the more the signal spreads out in other directions. The spread of the signal makes the signal weaker. When variation of the signal power occurs due to changes in the transmission medium or path, fading occurs. Basically, any interruption in the transmission medium (atmospheric changes) or path (objects) can affect the strength of the signal. Reflection exists when the signal bounces off large objects causing the signal to change. These changes can increase or decrease the signal strength. This usually happens when the signal reflects off walls, floors, or ceilings. Diffraction is produced when the signal runs into a large object. The secondary waves resulting from the obstructing surface are present throughout the space and behind the large object negatively affecting the strength of the transmitted signal. This can occur when the signal runs into a wall partition or cubicle. Scattering exists when the transmitted signal passes through many small objects that cause the signal to go in many different directions. Scattered waves are produced by rough surfaces, small objects, or by

other irregularities in the channel. Refraction is defined as a change in direction of a transmitted signal resulting from changes in velocity. This usually occurs when only part of the line of sight transmitted signal reaches the destination. Noise can be characterized as various distortions imposed by the transmission medium or additional unwanted signals. Noise is usually caused by interference or reception of unwanted signals from other electronic devices. Finally, a characteristic that affects all of these factors and their effect on the WiFi signal is the type of frequency of the network and interference (we assume 2.4GHz). A channel model is an equation that represents the path loss due to these factors. Due to the large number of obstacles that affect the strength of a transmitted WiFi signal, the channel models used to represent the environment must be very specific to each environment.

2.2 Existing Path Loss Models

The channel models that are discussed in this section include the Okumura-Hata model, Log-distance Path Loss model, and JTC Indoor Path Loss model. We will briefly describe each model and determine its suitability for the environment under consideration.

2.2.1 Okumura-Hata Model

The Okumura-Hata model is a combination of the Okumura model and empirical models developed by Masaharu Hata [9]. The Okumura-Hata model is represented below:

$$L_{50}(\text{urban})(dB) = 69.55 + 26.16 \log(f_c) - 13.82 \log(h_{te}) - a(h_{re}) + (44.9 - 0.55 \log(h_{te})) \log(d) \quad (1)$$

where L_{50} is the 50th percentile median path loss, f_c is the centre frequency in megahertz, h_{te} is the base station antenna height in meters, h_{re} is the receiver station antenna height in meters, $a(h_{re})$ is a vehicular station antenna height-gain correction factor that depends on the environment, and d is the link distance in kilometers [9, 10, 11].

The Okumura-Hata model is extremely accurate, because it is based on measurements in a specific environment. However, while the Okumura-Hata model is popular and accurate, it is mainly used in outdoor, urban environments [9, 10, 11]. The Okumura-Hata model would work well if we were determining path loss in network located outdoors. We should not use the Okumura-Hata model in the development of our signal strength monitoring system, because we are conducting measurements inside an office environment.

2.2.2 Log-distance Path Loss Model

The Log-distance Path Loss model is a very popular logarithmic model that is based on a linear dependence between the path loss in decibels and the logarithm of the distance between the transmitter and receiver [8, 9, 10, 11, 12]. This model predicts path loss inside a building or in densely populated areas. There also exist many studies that use a variation of the Log-distance Path Loss model [9, 11, 14, 15, 16, 18]. The Log-distance Path Loss model is represented below:

$$PL(d)(dB) = PL(d_0)(dB) + 10n \log(d/d_0) + X_\sigma \quad (2)$$

where $PL(d)(dB)$ is the measured path loss in decibels one meter from the transmitted signal, n is a path loss exponent dependant on the surroundings and building type, d is

the distance between the transmitter and receiver in meters, d_0 is typically one meter, and X_σ is a normal (Gaussian) random variable in decibels that has zero mean and standard deviation of decibels [8, 10, 14, 15]. This model also takes into consideration different obstacles in the transmitter to receiver path (also known as log normal shadowing).

Table 1 lists the path loss exponents based on different environments [8, 11, 18].

Environment	Path Loss Exponent, n
Free Space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

Table 1 - Log-distance Path Loss Exponent

According to many studies which used the Log-distance Path Loss model or a variation of this model, the Log-distance Path Loss model is accurate and simple to use [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. The Log-distance Path Loss model also will work in our environment and could be used in the development of our signal strength monitoring system.

2.2.3 JTC Indoor Path Loss Model

The JTC (Joint Technical Committee) Indoor Path Loss model is the official path loss model for office environments presented by the International Organization for Standardization (ISO). This model has traits from the Okumura-Hata model (based on specific measurements of different factors) and the Log-distance Path Loss model (based on the relationship between the logarithm of the distance between the transmitter and

receiver to the path loss in decibels). The JTC Indoor Path Loss model is represented below:

$$L_{total} = A + B \log_{10}(d) + L_f(n) + X_\sigma \quad (3)$$

where A is an environment dependent fixed loss factor in decibels, B is the distance dependent loss coefficient, d is the distance between the transmitter and receiver in meters, L_f is a floor/wall penetration loss factor in decibels, n is the number of floors/walls between the transmitter and receiver, and X_σ is a normal (Gaussian) random variable in decibels that has zero mean and standard deviation of σ decibels (log normal shadowing) [9, 11, 19]. Table 2 contains the corresponding variables dependent on the type of environment [9, 11, 19].

Environment	Residential	Office	Commercial
$A(dB)$	38	38	38
B	28	30	22
$L_f(n)(dB)$	4n	15 + 4 (n-1)	6 + 3 (n-1)
Log Normal Shadowing Std. Dev. (dB)	8	10	10

Table 2 - JTC Indoor Path Loss Model Variables

The JTC Indoor Path Loss model will work in our environment. According to [9], the JTC Indoor Path Loss model may be more accurate than the Log-distance Path Loss model due to the addition of the $L_f(n)$ function. The JTC Indoor Path Loss model could be used in the development of our signal strength monitoring system.

The Okumura-Hata, Log-distance Path Loss, and JTC Indoor Path Loss models are all accurate and reliable in different environments. Based on our environment of an indoor office setting, the Log-distance Path Loss and JTC Indoor Path Loss models could be used in the development of our signal strength monitoring system.

2.3 Development of Custom Channel Model

In order to develop our own channel model, we took many measurements in our custom environment and found the predicted value line based on our data. This predicted value line acts as a path-loss model in our environment. The area that we used for testing and developing our own channel model is a portion of the Information Assurance Research, Education, and Development Institute (IA-REDI) located on the sixth floor of the Marie V. McDemmond Center for Applied Research (MCAR) at Norfolk State University. This area is a computer lab (approximately twenty feet by sixty-five feet) next to an office, three conference rooms, and one long hallway. We used commercially available hardware and software. The access point we used was an Actiontec GT704WG router on default settings. The wireless card we used to connect to the access point was an Intel® PRO/Wireless 3945ABG Network Connection built-in a Dell Latitude D830 laptop running Windows XP on default settings.

The program we used to measure the signal strength is called inSSIDer (freeware) [20]. We measured the signal strength at eighteen locations on every hour between 9:00 a.m. and 5:00 p.m. (EST) for one week. Figure 9 displays the experiment area, measurement locations, and access point (AP). The access point is the solid circle in the upper right hand corner. The 'x' represents the measurement locations, and the arcs

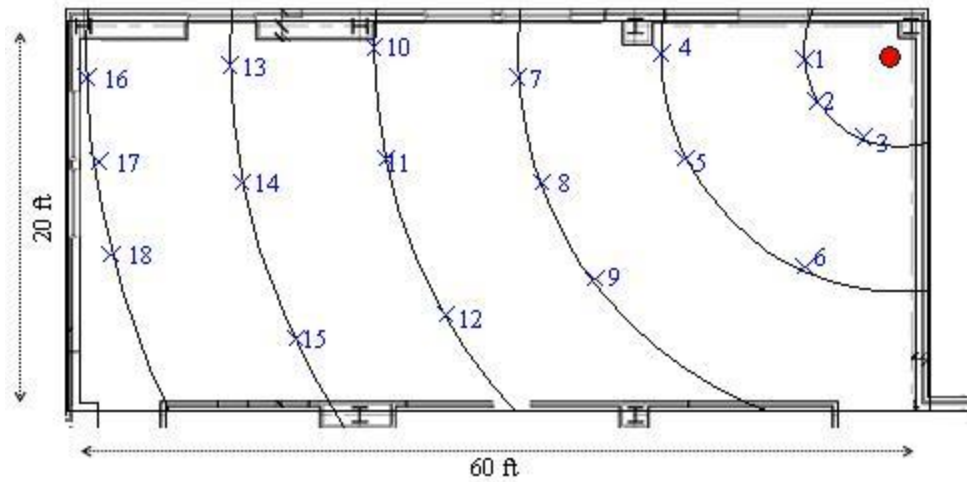


Figure 9 - Experiment Environment Overview

represent measured distances (ten feet) from the access point. Table 3 contains the measurement locations and their distance from the access point.

Location	Distance from AP (ft)	Distance from AP (m)
1	10	3.408
2	10	3.408
3	10	3.408
4	20	6.816
5	20	6.816
6	20	6.816
7	30	10.224
8	30	10.224
9	30	10.224
10	40	13.632
11	40	13.632
12	40	13.632
13	50	17.04
14	50	17.04
15	50	17.04
16	60	20.448
17	60	20.448
18	60	20.448

Table 3 - Distance of Locations from Access Point

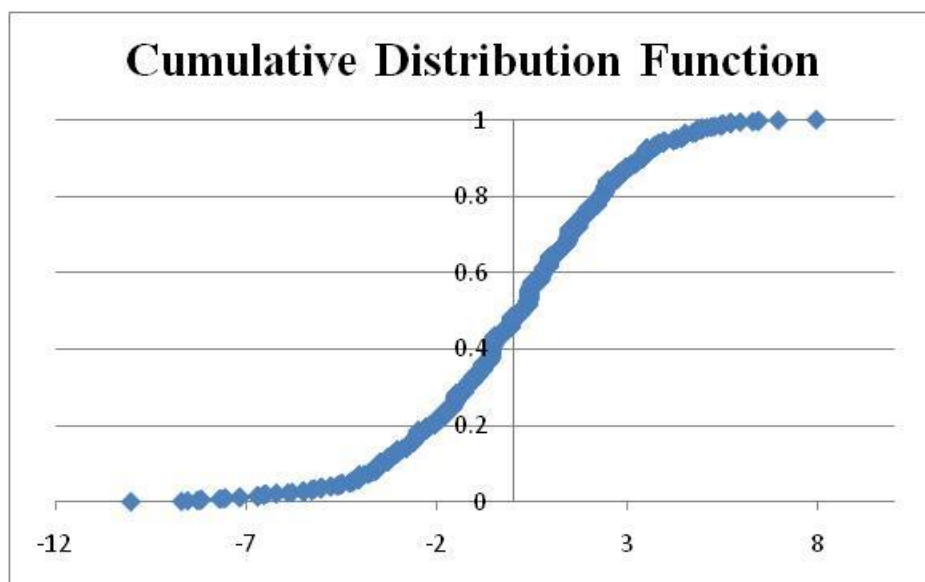


Figure 10 - Cumulative Distribution Function of Collected Data

The access point was sitting on a desk approximately three feet from the ground. When measuring the signal strength, the laptop was held approximately five feet from the ground with the screen facing away from the access point. The program inSSIDer was

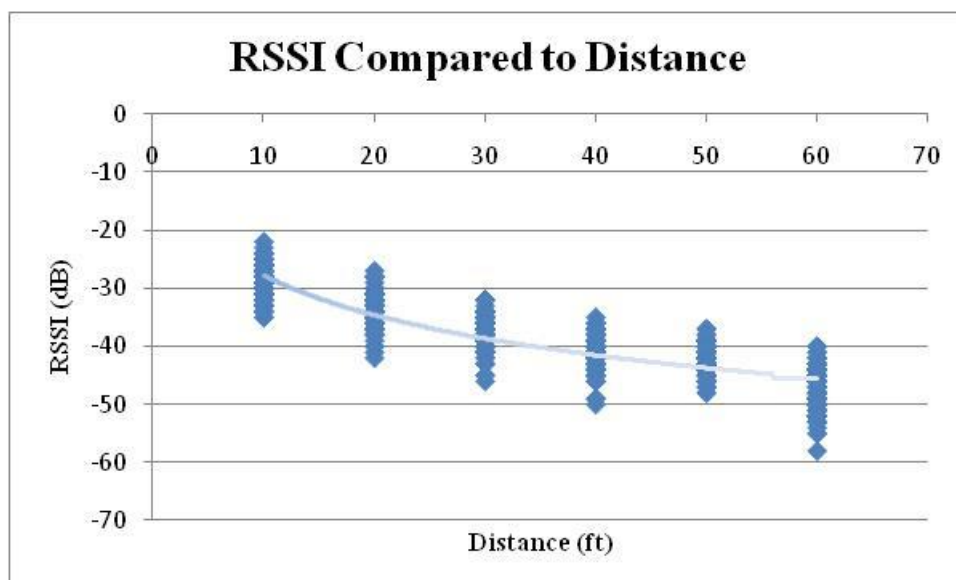


Figure 11 - Signal Strength Compared to Distance

used to collect data, and a total of 810 measurements were taken over a one-week time period.

Displayed in Figure 10 is the cumulative distribution function (CDF) of the measured signal strength versus distance. Figure 10 shows that the data collected (displayed in Figure 11) can be classified as normal.

Table 4 represents the average signal strength compared to distance from the access point. Figure 11 displays this relationship along with a predicted value line.

Distance from AP (ft)	Average SS (dB)	Variance
10	-28.82	7.55
20	-33.89	6.93
30	-37.31	7.4
40	-40.77	7.51
50	-42.44	5.14
60	-47.61	12.42

Table 4 - Average Signal Strength Compared to Distance

The equation for the predicted value line is:

$$PSS = (-2.1) \cdot 10 \log(d) - 7 \quad (4)$$

where is PSS the predicted signal strength and d is the distance from the access point in feet. This equation could serve as a channel model for this specific environment.

2.4 Selection of Channel Model

We identified three existing channel models and one new channel model to consider for implementation in this signal strength monitoring system. The new empirical model gives a path-loss exponent of about 2.1, which is closer to the free-space

path-loss exponent. In most experiments by other researchers, this value is much higher than the one we obtained. Since custom models can be more accurate in custom environments, a permanent, universal model may not be employed in designing signal mapping systems for 2.4 GHz IEEE 802.11 networks. Upon further review, we also conclude that one should not use the Okumura-Hata model, because we are conducting measurements inside an office environment. The Log-distance Path Loss model and JTC Indoor Path Loss model also will work in our environment, because they are tailored to be universal models for indoor use. However, an empirical model would be the most accurate to use in the same environment – as proposed in Figure 11.

3. SIGNAL STRENGTH MONITORING SYSTEM

3.1 Program Overview

In order to see if signal spilling exists in a network, we created a program to display the signal strength in our environment based on a channel model. We used C++, OpenGL, and GLUT. The GUI (graphical user interface) will display the environment in a two-dimensional setting with different colors representing different signal strengths. This program will show system administrators where the signal of their network is in a physical sense. If the signal of the network is still strong near the edges of the network, system administrators will know to weaken the strength of their network to prevent signal spilling. The system designed is referred to as the signal strength monitoring system (SSMS).

3.2 Static vs. Simulated Real-time vs. Real-time

We will give examples of a program that uses a static channel model. Static channel models are unchanging, but they include a random variable to simulate the randomness of a WiFi signal. For example, WiFi signals are constantly changing, and there may be many different signal strengths from the same location during different times of the day due to the obstacles discussed in Section 2.1. Static channel models are a good reference and are accurate, but they may not be realistic. In order to account for the small, continuous changes that occur in a live WiFi network, simulated real-time channel models generate the random variable and update it in real time (which updates the signal map in real time). This is the only difference between static and simulated real-time channel models. The simulated real-time channel model will update the random number every so many seconds. This ensures the map will be slightly different

every time the program updates. This is more realistic than using a static channel model. Even though a simulated real-time channel model is more realistic than a static channel model, the most accurate and realistic type of channel model for WiFi networks is real-time. In order to have a real-time channel model, a sensor grid will need to be deployed to constantly measure the signal strength of the network at various points. Every so often, a new channel model would be computed, and the signal would be re-displayed based on measurements. While real-time is the most accurate, we do not have a sensor grid to implement this type of model. In our environment, we have used a static channel model for examples (random variables are taken from a lognormal distribution).

3.3 Program Components

The major components used in the signal strength monitoring programs include: a struct to hold all of the information, draw functions to draw the environment and access point, a function to find the distance between each pixel and the access point, a function to calculate the signal strength based on a channel model and the calculated distance, a function to determine the color of the pixel based on the signal strength, a function used to color each pixel on the GUI, and a function that outputs the information to a text file.

3.3.1 Struct

A struct is used for each pixel located on the GUI. The struct includes an x-value (int xval), y-value (int yval), distance value (double d), and an RSSI value (double rssi). This information is kept for each pixel to maximize the efficiency of this program.

3.3.2 Draw Functions

Two draw functions are used to draw the environment and access point. The environment is draw using a line-loop in OpenGL. Essentially, the environment is

represented with a rectangle with static coordinates. The access point is drawn as a single point with a very large size. The coordinates for the access point are also static.

3.3.3 Finding the Distance

In order to find the distance between each pixel and the access point, the distance formula is used. In order to calculate the distance formula, the x-value and y-value of each pixel is compared to the x-value and y-value of the access point. As noted earlier, the value for distance is stored for each pixel in the struct.

3.3.4 Calculate Signal Strength

In order to calculate the signal strength for each pixel, the distance value is plugged into the channel model. The channel model we will use is our custom channel model with the addition of the random number generator. Both channel models are represented in their appropriate programs. As mentioned earlier, the value for signal strength is stored in the struct.

3.3.5 Signal Strength to Color Conversion

Each pixel in the environment area will be colored based on its signal strength. Using different colors makes it easier to visualize the different signal strengths in the environment. Table 5 represents the conversion made from signal strength to color.

3.3.6 Color Each Pixel

A point is drawn on each pixel based on the color from the signal strength to color conversion, x-value, and y-value. Only pixels inside of our environment are colored. It would not be appropriate to color pixels outside of our environment, because these locations were not considered in the development of our channel model.

SS (dB)	Color Name	Color	glColor3ub(R,G,B)
0 to -10	Red		255, 0, 0
-11 to -20	Light Red		255, 110, 0
-21 to -30	Orange		255, 175, 0
-31 to -40	Orange Yellow		255, 225, 0
-41 to -50	Yellow		255, 255, 0
-51 to -60	Yellowish Green		175, 255, 0
-61 to -70	Green		0, 255, 0
-71 to -80	Baby Blue		0, 255, 255
-81 to -90	Light Blue		0, 175, 255
-91 to -100	Dark Blue		0, 0, 255

Table 5 - Signal Strength to Color Conversion

3.3.7 Output Information to File

The last action the program takes is outputting the information inside of the struct to a text file. The purpose of this action is to log the data at a given time and make it available for use by other people/programs if necessary.

3.4 Output Examples

The first example is from main2.cpp that has our static custom channel model with no randomness. Figure 12 represents the output from main2.cpp. As seen in Figure 12, the circles that represent the different signal strengths are symmetric. This is accurate, because it employs our custom channel model. However, this is not realistic, because the signal strength in a live environment will not be symmetric.



Figure 12 - Static Custom Channel Model (no randomness) Output

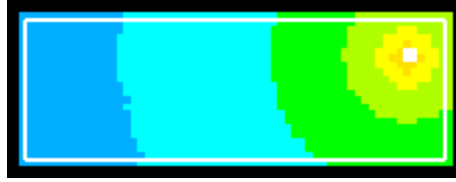


Figure 13 - JTC Indoor Path Loss Model (with randomness) Output

The second example is from main3.cpp that has the JTC Indoor Path Loss Model. This channel model is static, but it includes the random number generator. Figure 13 represents the output from main3.cpp. As you can see, the circles that represent the different signal strengths are not symmetric. This is realistic, because the WiFi signal will never be symmetric. However, the colors pictured in Figure 13 are much different from the colors pictured in Figure 12. According to the JTC Indoor Path Loss Model, the furthest distances away from the access point will be between -81 and -90 decibels. However, based on our measurements the same area should be between -41 and -50 decibels. This discrepancy is very large and could cause problems for network administrators if they choose the wrong model. This is an example of how different channel models can give different results in the same environment.

The third example is from main4.cpp that has our static custom channel model with randomness. Figure 14 represents the output from main4.cpp. As seen in Figure 14, this output includes accurate data and asymmetric circles. This is the most accurate and



Figure 14 - Static Custom Channel Model (with randomness) Output

realistic output we can have without deploying a sensor grid for a real-time SSMS or constantly updating program.

3.5 Results

The programs discussed in this section accurately depict the signal strength of a WiFi network in a custom environment. If the same type of system were to be deployed in a different environment, a channel model custom to that environment should be used for the most accurate results. In order to be the most accurate and realistic, a real-time system should be used. If a real-time system is not available, a simulated real-time system should be used. This system will allow system administrators to view where the WiFi signal is on a physical level. This system will allow system administrators to make a decision to limit power if signal spilling is occurring. The use of our SSMS can lead to the prevention of signal spilling on WiFi networks.

4. LOCATION DETERMINATION SYSTEM

4.1 Geolocation Models

The geolocation models that are discussed include Global Positioning System (GPS), Angle of Arrival (AOA), Time-based models, Ahmad's Algorithm for Closest Vertices, and a Signal Strength-based model (SS). We will briefly describe each model and determine if it can be used in our environment. Later, we will pick the most accurate geolocation model for our specific environment. This geolocation model will act as our location determination system (LDS) for WLANs.

4.1.1 GPS

One of the most accurate geolocation systems in use is GPS. As stated in [21, 22], GPS consists of a constellation of twenty-four satellites (synchronized), equally spaced in six orbital planes 20,200 kilometers above the earth. Figure 15 represents the GPS satellite constellation. GPS receivers are used to calculate their exact position (longitude, latitude, and altitude) based on measured signals from at least four (must be able to have line-of-sight to at least four satellites) of these twenty-four satellites. In order to calculate the GPS receiver's location, the GPS receiver compares the time the messages are sent and the satellites' locations. In terms of accuracy, GPS can be exact to around ten meters [21, 23, 24].

The first shortcoming of implementing GPS concerns calculation time. If the GPS receiver starts without any knowledge of the GPS constellation's state, it may take as long as several minutes for locations to be calculated [21]. Also, as mentioned in [21, 22, 23, 24], in order for GPS to operate properly, the GPS receiver needs to have line-of-sight to at least four GPS satellites. If the GPS receiver cannot connect to at least four

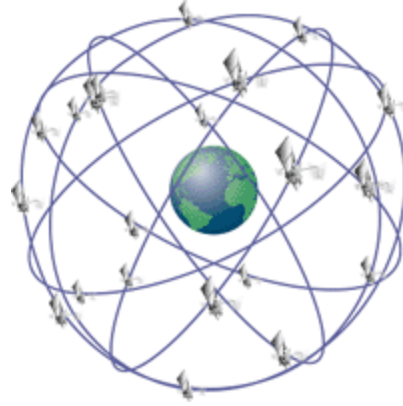


Figure 15 - GPS Satellite Constellation [25]

satellites, this system will not work at all. Therefore, GPS will not work indoors [21, 23, 24, 26].

4.1.2 AOA

AOA geolocation systems use antenna arrays and the angle of the array from the client to multiple base stations to calculate specific locations [22, 23, 27, 28]. Figure 16 represents how the measurements of antenna arrays can be calculated into location (based on [22]). Node C represents a client connected to the sensor network. Nodes S represent

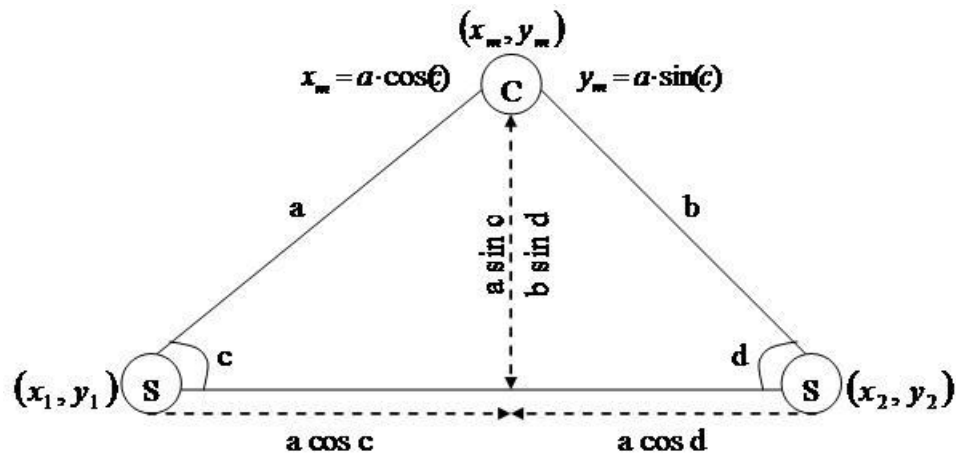


Figure 16 - AOA Measurements and Calculations [23]

sensors, and all nodes have x-y coordinates. The distance between node C and nodes S are represented by (a) and (b). The angle between the antenna arrays and sensor nodes are represented by (c) and (d).

According to [22, 26, 27, 28, 29], AOA geolocation systems have accuracy issues indoors due to multipath interference. In order for the measurements to be extremely accurate, line-of-sight is required from the client to the sensors. If there is no line-of-sight, measurements will not be accurate.

4.1.3 Time-based Models

There are two types of time-based geolocation systems; Time of Arrival (TOA) and Time Difference of Arrival (TDOA). As stated in [26], TOA geolocation systems measure distance based on an element of propagation delay between a transmitter and a receiver since in free space or air, radio signals travel at the constant speed of light. In order for the calculations to be exact, the internal clocks of the sensors and client need to be synchronized. There are many equations used to calculate the distance estimates as discussed in [22, 23, 26, 27, 28]. TDOA takes the formulas used in TOA and adds more estimation in order to account for the lack of synchronization between the client and sensor nodes. TDOA still requires the sensors' clocks to be synchronized. In more detail, the TDOA of two signals traveling between the client and two sensors is estimated, which determines the location of the client on a hyperbola, with foci at the two reference nodes (a third sensor is used for localization) [27, 28, 30].

One of the shortcomings for TOA is the requirement for the sensors and client to have synchronized clocks [22, 23, 26, 27, 28, 30]. If the client or sensors are not synchronized, the TOA output will be inaccurate. Even though TDOA does not require

synchronization between the client and sensors, it still has accuracy issues due to the estimation of clock delay between the client and sensors [22, 23, 27, 28, 30]. If the estimate for TOA between the client and sensors is not accurate, the location computed in the TDOA calculation will not be accurate. Finally, as with AOA geolocation systems, TOA and TDOA will perform poorly if there is no line-of-sight [27, 28, 29]. Multipath interference will significantly reduce the accuracy of TOA and TDOA geolocation systems.

4.1.4 Ahmad's Algorithm of Closest Vertices³

This algorithm compares the signal strength values for the wireless station received at all of the sensors in the grid. The objective is to determine the quadrant (or triangle) where the client is located. In more detail, a list is compiled of the signal strength of the client at each sensor. Next, the list is sorted from the strongest signal to the weakest. Finally, the algorithm selects which quadrant the client is in based on four rules:

- i. if the four strongest signal strength nodes form one quadrant, the client is located in that quadrant,
- ii. if the three strongest signal strength nodes are from one quadrant, the client is located in that quadrant,
- iii. if the two strongest signal strength nodes are from one quadrant and they form a vertical line, two neighbors (left and right) of one of the nodes are compared where the strongest signal strength results in the quadrant where the client is located,

³ Due to Professor Aftab Ahmad of Computer Science Department, Norfolk State University

- iv. if the two strongest signal strength nodes are from one quadrant and they form a horizontal line, two neighbors (above and below) of one of the nodes are compared where the strongest signal strength results in the quadrant where the client is located.

Figure 17 represents a sensor grid with quadrants to further explain Ahmad's Algorithm for Closest Vertices. The sensors are denoted with labeled squares, and the circle is the client. The dashed lines form a quadrant. The following example refers to Figure 17. As rule one stipulates, if the four strongest signal strengths are for nodes F, G, J, and K, the client is located in quadrant FGJK. As rule two stipulates, if the three strongest signal strengths are for nodes F, G, and K, the client is located in quadrant FGJK. As rule three stipulates, if the two strongest nodes are G and K, the signal strength values for nodes F and H or J and L are compared. If nodes F or J have a stronger signal strength value for the client than H or L, the client is located in quadrant FGJK. Otherwise, the client is located in quadrant GHKL. As rule four stipulates, if the two strongest nodes are G and F, the signal strength values for nodes C and K or B and J are compared. If nodes C or B have a stronger signal strength value for the client than J or K, the client is located in quadrant BCFG. Otherwise, the client is located in quadrant FGJK.

Ahmad's Algorithm for Closest Vertices is simple and convenient for networks with sensor grids, but the efficiency can depend on the size of the quadrants. For example, if the quadrants are very small, mobile clients will be changing quadrants constantly. If the client moves randomly while attacking a network with this design and the quadrants are small, the quadrant where the client is located will change too quickly

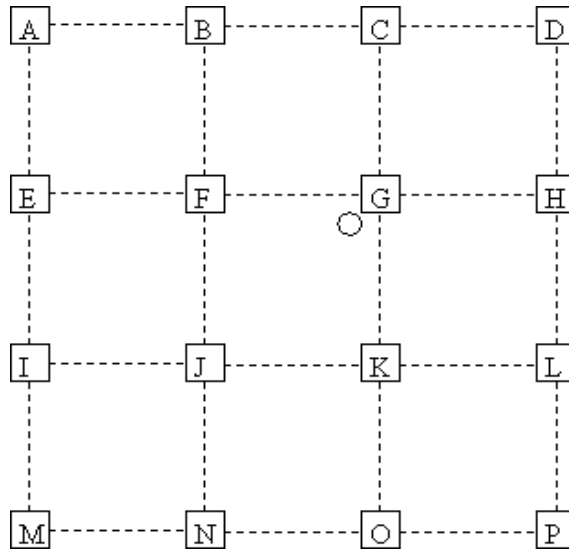


Figure 17 - Sensor Grid with Quadrants

to provide a reliable location. Also, if the quadrants are very large, it will take more time to search a quadrant for a specific client. Furthermore, if a client were to walk around node G in Figure 17, quadrants BCFG, CDGH, FGJK, and GHKL would need to be searched. There have been no implementations of Ahmad's Algorithm for Closest Vertices yet to test the ideal size of quadrants.

4.1.5 SS based

SS based geolocation systems do not rely on line-of-sight. SS systems are well suited for indoor use, because they account for multipath interference [23, 27, 28, 30]. The combination of measured signal strength and a path loss model will produce a value that represents the distance between the client and sensor [23, 27, 28, 29, 30]. In more detail, a channel model represents the path loss a signal will experience in a given transmission medium. The signal strength of a client compared to a sensor can be entered into a channel model to produce a distance. If this calculation is completed between one client and three sensors, the client's location can be determined. Figure 18 represents the

SS location determination process. The client connected to the network is represented by (C). There are three sensors (Sa), (Sb), and (Sc) that have signal strength values for the client. These signal strength values are plugged into the channel model to get the distance the client is from each sensor (Da), (Db), (Dc). Next, circles are drawn to represent the points where the client could possibly be located. Finally, the intersection of all three circles represents the client's physical location.

While SS geolocation systems eliminate the need for line-of-sight, estimation still occurs due to multipath interference of the radio signal. As stated in [27, 28, 30], a channel model is needed to predict the path loss in a given medium. This channel model estimates the relationship between distance and signal strength. Therefore, the accuracy

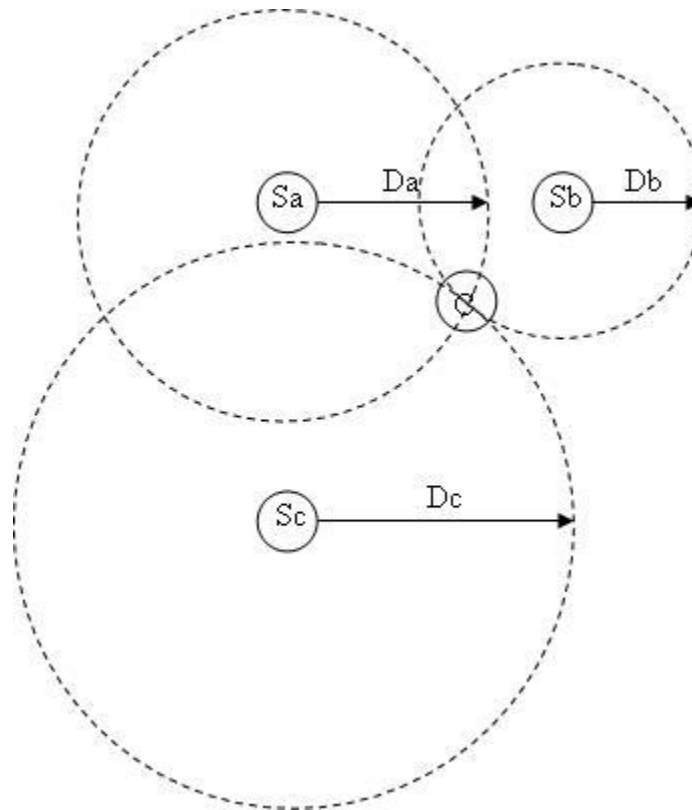


Figure 18 - SS Location Determination Process [22]

of SS geolocation systems depends on the exactness of the channel model used in the SS calculation process. If the channel model is not accurate, the results of the SS calculation process will not be accurate.

4.2 Selection of Geolocation Model for LDS

Based on the shortcomings of GPS previously listed, GPS is not a system that would work for our LDS for WLANs in our environment. First, GPS can take several minutes to calculate locations. Our LDS for WLANs will not be able to wait several minutes while locations are being calculated. Our LDS for WLANs must be able to calculate locations in a few seconds. Second, GPS will only work with line-of-sight to at least four satellites. Our system will be deployed indoors, so no line-of-sight is possible. GPS is not a system we will use for our LDS for WLANs.

Due to the fact AOA measurements are not accurate all of the time, AOA is not a system that would work for our LDS for WLANs. Our system is located in an indoor environment that is deployed in a multipath channel. The multipath interference will cause AOA measurements to not be accurate all of the time. While it is possible AOA will work some of the time, we need a LDS for WLANs that will be accurate on a consistent basis. AOA is not a system we will use for our LDS for WLANs.

As reported previously, TOA requires synchronization between sensors and client. While it is acceptable to assume the sensors will be synchronized in our sensor grid, it is not acceptable to assume clients will be synchronized with the sensors. In order to eliminate the need of synchronization, TDOA makes estimates about the difference in specific TOA values. However, if these estimates are not accurate, the results of the TDOA calculation will not be accurate. We would rather employ a geolocation system

that does not depend on estimates. Lastly, TOA and TDOA will not be consistently accurate in our environment, because our environment has a multipath channel. Due to synchronization requirements, calculations based on estimates, and inadequate resistance to multipath interference, we will not implement TOA or TDOA geolocation systems for our LDS for WLANs.

Even though Ahmad's Algorithm for Closest Vertices is straightforward and accommodating for networks with sensor grids, there are still many questions about it that needs answering. For example, the ideal quadrant size needs to be determined along with the performance and computational requirements. After these characteristics are known, there will be enough information to determine whether this algorithm is appropriate for use for a LDS for WLANs. At this point in time, we will not use Ahmad's Algorithm for Closest Vertices for our LDS for WLANs, because we do not know how it would perform (accuracy and speed) against other geolocation systems.

In our environment, we measured the signal strength in various locations throughout a working week in order to calculate an accurate channel model. Our channel model is accurate, because it is based on actual measurements. If we use our simulated real-time custom channel model in the SS calculation process, the results would be extremely accurate. Also, this would require no additional equipment other than the sensor grid we would already have in place. Until the writing of this paper, the SS geolocation system in combination with our simulated real-time custom channel model is identified to be the best LDS for WLANs.

4.3 Results

The SS geolocation system in combination with our simulated real-time custom channel model would be an extremely accurate LDS for WLANs. Unfortunately, we were never able to acquire enough sensors to deploy a sensor grid to empirically test the geolocation models discussed earlier. When this system is implemented with a sensor grid, it will allow system administrators to physically see where all of the clients are located on their network, including attackers. This will allow system administrators to make quicker and better decisions when responding to attacks on their network. The use of our LDS for WLANs can lead to the prevention of attacker anonymity on WiFi networks.

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

Upon completing a background study comparing the security of WLANs to the security of wired LANs, two specific problems exist at the physical layer in WiFi. First, attacks are anonymous at the physical layer, because there are no physical connections. With wired LANs, one can physically trace each packet's source and destination machine using commonly available tools, and by running cable through designated areas, thus providing strong privacy. However, there is no way to physically trace a packet on a WiFi network to see the physical location of the source or destination machine. An attacker can attack a network anywhere a WiFi signal is detected. Second, the WiFi signal is uncontrolled. There is no way to control or view the WiFi signal according to the IEEE 802.11-2007 protocol. Directional antennas restrict the service availability but will not provide signal containment. Signal spilling can occur when a WiFi signal is transmitted further than intended. This makes it possible for attacks to occur outside of the physical building where the WiFi network is located. These two problems lead to WiFi being very insecure at the physical layer.

In order to combat signal spilling, a signal strength monitoring system was designed using a simulated real-time custom channel model. When the SSMS is used, system administrators can view the strength of the WiFi signal on their network to see if signal spilling is occurring. If signal spilling exists, system administrators can make the decision to reduce the power of their network to try to contain the WiFi signal. The use of our proposed SSMS can lead to the prevention of signal spilling.

In order to prevent attackers from being anonymous, a geolocation model was combined with our simulated real-time custom channel model to produce our location determination system. When our LDS is used, all of the clients connected to a WiFi network will be physically plotted on a map and put in an identification table (IP address, MAC address, or other form of identification). If an attack were to occur, system administrators would just need to match the attacking client to their plot in order to deduce the attacker's physical location.

The combination of our SSMS and LDS make WiFi networks much more secure on the physical layer of the internet protocol.

5.2 Future Work

In order to test the SSMS and LDS in more detail, a WiFi sensor grid could be implemented in our custom environment. This would allow us to empirically compare geolocation models to provide more specific comparisons and results. Also, we would be able to test Ahmad's Algorithm of Closest Vertices to determine an ideal quadrant size, its performance, and power consumption. In order to set up a WiFi sensor network, low-power sensors need to be acquired. Finally, after the implementation of a WiFi sensor network with low-power sensors, work can begin on feedback power control for access points. Currently, our SSMS identifies signal spilling. To make our SSMS complete, we could add a power control feature that would allow system administrators to control the power of their access points. This would extend the function of our SSMS from identifying signal spilling to preventing signal spilling.

REFERENCES

- [1] IEEE. "Std 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Section 8--Security".
- [2] Zahur, Y. and T. A. Yang. "Wireless LAN Security and Laboratory Designs". Consortium for Computing Sciences in Colleges, pp. 44-60, 2003
- [3] Sriram, V. S. S. and G. Sahoo. "Securing IEEE 802.11 Wireless LANs - A Mobile Agent Based Architecture". In Proceedings of IEEE International Advance Computing Conference. Patiala, India, pp. 1187-1191, 2009.
- [4] He, C., et al. "A Modular Correctness Proof of IEEE 802.11i and TLS". In Proceedings of ACM Conference on Computer and Communications Security. Alexandria, Virginia, USA, pp. 2-15, 2005.
- [5] He, C. and J. C. Mitchell. "Analysis of the 802.11i 4-Way Handshake". In Proceedings of ACM Workshop on Wireless Security. Philadelphia, Pennsylvania, USA, pp. 43-50, 2004.
- [6] Brustoloni, J. C. "Laboratory Experiments for Network Security Instruction". ACM Journal on Educational Resources in Computing (n.d.), pp. 1-18.
- [7] Hori, Y. and K. Sakurai. "Security Analysis of MIS Protocol on Wireless LAN comparison with IEEE802.11i". Mobile Technology, Applications, and Systems. Bangkok, Thailand, pp. 1-5, 2006.
- [8] Kent, S. and K. Seo. "Security Architecture for the Internet Protocol". December 2005.
- [9] Tummala, D. "Indoor Propagation Modeling at 2.4 GHz for IEEE 802.11 Networks". M.S. Thesis, University of North Texas, 2005.
- [10] Pahlavan, K., and Levesque, A. H. "Wireless Information Networks". Wiley-Interscience. New York, New York, pp. 73-112, 1995.
- [11] Vig, J. "ISM Band Indoor Wireless Channel Amplitude Characteristics: Path Loss vs. Distance and Amplitude vs. Frequency". M.S. Thesis, Ohio University, 2004.
- [12] Tipper, D. "Wireless Communication Fundamentals". University of Pittsburgh lecture, pp. 40-42, 2005.
- [13] Faria, D. B. "Modeling Signal Attenuation in IEEE 802.11 Wireless LANs". Stanford University, July 2005.

- [14] Akl, R., Tummala, D., and Li, X. "Indoor Propagation Modeling at 2.4 GHz for IEEE 802.11 Networks". In Proceedings of the 6th IASTED International Multi-Conference on Wireless and Optical Communications, Banf, AB, Canada, 2006.
- [15] Borrelli, A., et al. "Channel Models for IEEE 802.11b Indoor System Design". In Proceedings of IEEE Conference on Communications, vol. 6, pp. 3701-3705, 2004.
- [16] Andrade, C. B. and Hoeful, R. P. F.. "IEEE 802.11 WLANs: A Comparison on Indoor Coverage Models". In Proceedings of the 23rd Canadian Conference on Electrical and Computer Engineering, 2010.
- [17] Capulli, F., et al. "Path Loss Models for IEEE 802.11a Wireless Local Area Networks". In Proceedings of the 3rd International Symposium on Wireless Communications Systems, 2005.
- [18] Liechty, L. "Path Loss Measurements and Model Analysis of a 2.4 GHz Wireless Network in an Outdoor Environment". M.S. Thesis, Georgia Institute of Technology, 2007.
- [19] Phaiboon, S. "An Empirically Based Path Loss Model for Indoor Wireless Channels in Laboratory Building". In Proceedings of the IEEE TENCON'02, 2002.
- [20] Joint Technical Committee of Committee T1 R1P1.4 and TIA TR46.3.3/TR45.4.4 on Wireless Access, "Draft Final Report on RF Characterization," PaperNo. JTC(AIR)/94.01.17-238R4, Jan. 17, 1994.
- [21] <http://www.metageek.net/products/inssider>
- [22] Djuknic, G. M., and Richton, R. E. "Geolocation and Assisted GPS". Computer, vol. 24, no. 2, pp. 123-125, Feb. 2001.
- [23] Sayed, A. H., Tarighat, A., and Khajehnouri, N. "Network-Based Wireless Location: Challenges Faced in Developing Techniques for Accurate Wireless Location Information". Signal Processing Magazine, IEEE, vol. 22, no. 4, pp. 24-40, July 2005.
- [24] Gustafsson, F. and Gunnarsson, F. "Mobile Positioning Using Wireless Networks: Possibilities and Fundamental Limitations Based on Available Wireless Network Measurements". Signal Processing Magazine, IEEE, vol. 22, no. 4, pp. 41-53, July 2005.
- [25] Xiujun Li, Gang Sun, and Xu Wang. "Mobile Positioning System Based on the Wireless Sensor Network in Buildings". In Proceedings of the 5th International

Conference on Wireless Communications, Networking and Mobile Computing, pp. 96-100, Sept. 2009.

- [26] http://www.declarepeace.org.uk/captain/murder_inc/site/911-7.html
- [27] Pahlavan, K., Xinrong Li, and Makela, J. P. "Indoor Geolocation Science and Technology". *Communications Magazine*, IEEE, vol. 40, no. 2, pp. 112-118, Feb. 2002.
- [28] Gezici, S. "A Survey on Wireless Position Estimation". *Wireless Personal Communications*, vol. 44, no. 3, pp. 263-282, Feb. 2008.
- [29] Gezici, S., et al. "Localization via Ultra-Wideband Radios: A Look at Positioning Aspects for Future Sensor Networks". *Signal Processing Magazine*, IEEE, vol. 22, no. 4, pp. 70-84, July 2005.
- [30] Tsung-Nan Lin and Po-Chiang Lin. "Performance Comparison of Indoor Positioning Techniques Based on Location Fingerprinting in Wireless Networks". In *Proceedings of the IEEE TENCON'02*, 2002.
- [31] Yihong Qi. "Wireless Geolocation in a Non-Line-of-Sight Environment". Ph.D. dissertation, Princeton University, Nov. 2003.