

Louisiana Tech University

Louisiana Tech Digital Commons

Doctoral Dissertations

Graduate School

Fall 8-2020

0E2FA: Zero Effort Two-Factor Authentication

Ali Abdullah S. AlQahtani

Follow this and additional works at: <https://digitalcommons.latech.edu/dissertations>

**0E2FA: ZERO EFFORT TWO-FACTOR
AUTHENTICATION**

by

Ali Abdullah S. AlQahtani, B.S., M.S., M.S.E.

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy: Cyberspace Engineering

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

August 2020

LOUISIANA TECH UNIVERSITY
GRADUATE SCHOOL

June 22, 2020

Date of dissertation defense

We hereby recommend that the dissertation prepared by

Ali Abdullah S. AIQahtani, B.S, M.S., M.S.E.

entitled **0E2FA: Zero Effort Two-Factor Authentication**

be accepted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Engineering, Cyberspace Conc.

Jean Gourd
Supervisor of Dissertation Research

Jean Gourd
Head of Engineering

Doctoral Committee Members:

Manki Min
Miguel Gates
Jinyuan Chen
Ibrahim Abdoulahi

Approved:

Hisham Hegab
Dean of Engineering & Science

Approved:

Ramu Ramachandran
Dean of the Graduate School

ABSTRACT

Smart devices (mobile devices, laptops, tablets, etc.) can receive signals from different radio frequency devices that are within range. As these devices move between networks (e.g., Wi-Fi hotspots, cellphone towers, etc.), they receive broadcast messages from access points, some of which can be used to collect useful information. This information can be utilized in a variety of ways, such as to establish a connection, to share information, to locate devices, and to identify users, which is central to this dissertation. The principal benefit of a broadcast message is that smart devices can read and process the embedded information without first being connected to the corresponding network. Moreover, broadcast messages can be received only within the range of the wireless access point that sends the broadcast, thus inherently limiting access to only those devices in close physical proximity, which may facilitate many applications that are dependent on proximity. In our research, we utilize data contained in these broadcast messages to implement a two-factor authentication (2FA) system that, unlike existing methods, does not require any extra effort on the part of the users of the system. By determining if two devices are in the same physical location and sufficiently close to each other, we can ensure that they belong to the same user. This system depends on something that a user *knows*, something that a user *owns*, and—a significant contribution of this work—something that is *in the user's environment*.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that “proper request” consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author _____

Date _____

DEDICATION

I dedicate my dissertation to my first teacher and best friend in the world, my mother, Shahrah Saeed. She is the person who I ran to when I needed help, support, and comfort. As a child, I would only seek her for protection and happiness. A simple kiss from my mother allows me to forget all my worries and troubles. Her words of encouragement and lessons of patience has allowed me to reach this level of education. Without your love and support, I would not be the man that I am today. You have become my eyes that I see with, my ears that I hear with, and my mind that I think with. Your influence in my life is beyond measurable. Without you, I am the result of multiplying by zero.

To my father, who left this world but still lives inside me. Your death while I was a teenager created a huge hole in my life. Not a day goes by without me thinking about you. I wish that I could hug you one last time. Losing you so early in life forced me to adapt quickly and learn how to be man at an early age. Even in your absence you benefitted me with remarkable fortitude to accomplish whatever goal I set. I will always be grateful to you father. May Allah bless you and have mercy on your soul.

I have failed countless times during this journey. However, I strived to learn from these failures and used it as inspiration to find a new path to reach the finish line. The purpose was not for self-fulfillment and boastfulness. Rather, it was for the entire world to know I am the son of great parents in this life. I am the son of Abdullah and Shahrah.

To my children: I want you to be better than me. I want to be nothing compared to you. Make me proud to say to the whole world that you are my children.

Failing is a path to find success.

Yes, I might be an educated person but

when it comes to my mom, I am a child,

when it comes to my siblings, I am a brother,

when it comes to my wife, I am a husband,

when it comes to my kids, I am a kid too,

and when it comes to myself, I am Ali

so, I love to live my life as a normal, humble person.

TABLE OF CONTENTS

ABSTRACT.....	iii
DEDICATION.....	v
LIST OF TABLES.....	xii
LIST OF FIGURES.....	xiii
ACKNOWLEDGMENTS.....	xv
CHAPTER 1 INTRODUCTION.....	1
1.1 Authentication Credentials.....	2
1.1.1 Something a User Knows.....	2
1.1.2 Attacks on Passwords.....	3
1.1.2.1 Social engineering.....	3
1.1.2.2 Capturing a password.....	4
1.1.2.3 Brute force attack.....	4
1.1.2.4 Dictionary attack.....	5
1.1.2.5 Hybrid attack.....	5
1.1.2.6 Rainbow tables.....	5
1.1.3 Something a User Possesses.....	6
1.1.3.1 Tokens.....	6
1.1.3.2 Smart cards.....	7
1.1.3.3 Mobile devices.....	8

1.1.4 Something a User Is	8
1.1.4.1 Standard biometric authentication	8
1.1.4.2 Cognitive biometric authentication.....	9
1.1.5 Something a User Does.....	10
1.1.5.1 Keystroke dynamics.....	10
1.1.5.2 Voice recognition.....	11
1.1.6 Somewhere a User Is	12
1.2 Technical Background	13
1.3 Contribution	21
1.4 Guide to Dissertation and Conclusion	21
1.5 Peer Review	22
CHAPTER 2 BACKGROUND	23
2.1 Related Works.....	23
2.1.1 Authentication Systems that Depend on a OTP.....	23
2.1.2 Authentication Systems that Depend on a Phone Call.....	25
2.1.3 Authentication Systems that Depend on RFID Technology.....	28
2.1.4 Authentication Systems that Depend on QR Technology	30
2.1.5 Authentication Systems that Depend on NFC Technology	32
2.1.6 Authentication Systems that Depend on Ambient Sound.....	34
2.2 Conclusion	35
CHAPTER 3 THE PROPOSED SYSTEM	37
3.1 The Proposed System’s Architecture.....	37
3.2 The Proposed System’s Design Specifics.....	39

3.2.1 Requiring Zero-Effort to Implement Second Layer of Authentication	39
3.2.2 Verification Processes.....	40
3.2.3 The Proposed System’s Components.....	41
3.2.4 Database Design.....	42
3.2.5 Determining the Location of a User’s Devices.....	45
3.3 Operation.....	46
3.3.1 Communication Overview	49
3.3.2 Implementation	52
3.4 The Proposed System’s Features	58
3.4.1 Security	58
3.4.2 User Convenience	59
3.4.3 Flexibility.....	59
3.4.4 Cost-Effectiveness	59
3.4.5 Logging in Using a Login Machine	60
3.5 Challenges.....	60
3.5.1 Collecting Data from a User’s Mobile Device	60
3.5.2 Collecting Data from a User’s Login Machine (or Similar) Computer	61
3.5.3 Gaining Access	61
3.6 Conclusion	62
CHAPTER 4 DISCUSSION.....	64
4.1 Security	64
4.1.1 Security Features.....	65
4.1.1.1 The client-side application security	65

4.1.1.2	Limitation of spoofing	66
4.1.1.3	Communication security	66
4.1.1.4	Parameter adjustability.....	68
4.1.1.5	Database and infrastructure security.....	68
4.1.1.6	Strong second layer of authentication.....	68
4.1.2	Cyberattack Vulnerability Assessment	69
4.1.2.1	Man-in-the-middle attack.....	70
4.1.2.2	Eavesdropping.....	70
4.1.2.3	Simulation of the user's environment.....	71
4.1.2.4	An Attacker inside a user's environment.....	72
4.2	User Convenience	72
4.3	User Privacy.....	73
4.4	Robustness	74
4.4.1	Lack of Access Points in the User's Environment.....	74
4.4.2	Absence of the User's Mobile Device	74
4.5	Cost.....	76
4.6	Example Scenario	76
4.6.1	Session Termination.....	77
4.7	Conclusion	79
CHAPTER 5 EXPERIMENTS.....		80
5.1	Experiment 1: Testing Location Accuracy in a Predefined Area	82
5.1.1	Calculating the User Device Location Using a Single RSSI Reading.....	84
5.1.2	Calculating Location Using the Average of Thirty RSSI Readings	85

5.1.3 Calculating Location Using the Median of Thirty RSSI Readings.....	86
5.2 Experiment 2: Testing the Authentication Success Rate	88
5.2.1 Random Forest Model.....	90
5.2.2 k-Nearest Neighbors (k-NN) Model	91
5.2.3 Decision Tree Model.....	94
5.2.4 Random Tree Model	96
5.3 Experiment 3: Testing the RSSI Behaving Using Different Phone Models	97
5.4 Experiment 4: Testing the Computing and Communication Cost	98
5.5 Experiment 5: Testing the Proposed System’s Robustness Through Implementation	100
5.6 Experiment 6: Lack of Access Points	102
5.7 Experiment 7: Mobile Device Application Security.....	104
5.8 Experiment 8: Battery Consumption	105
5.9 Conclusion	106
CHAPTER 6 CONCLUSIONS AND FUTURE WORK.....	108
6.1 Conclusions.....	108
6.2 Future Work	110
6.2.1 Collecting Wi-Fi Footprints Through the Login Machine.....	110
6.2.2 Continuous Authentication	110
6.2.3 Estimating the Distance Between the User’s Devices	111
APPENDIX A SOURCE CODE	112
REFERENCES	120

LIST OF TABLES

Table 5-1: The calculated location using a single RSSI reading	85
Table 5-2: The calculated location using the average of thirty RSSI readings.....	86
Table 5-3: The calculated location using the median of thirty RSSI readings.	87
Table 5-4: Random forest model result.....	91
Table 5-5: k-nearest neighbors (k-NN) model result	93
Table 5-6: Decision tree model result	95
Table 5-7: Random tree model result.....	96
Table 5-8: RSSI measurements.....	98
Table 5-9: Duration of time for 50 login attempts in seconds	99
Table 5-10: Result using one access point	103

LIST OF FIGURES

Figure 1-1: 802.11 Beacon frame [16].....	15
Figure 1-2: A list of access points showing the SSID and signal strength of each	18
Figure 1-3: List of SSIDs (Network Name column) with their respective BSSIDs.....	19
Figure 1-4: Beacon frame information for several access points	20
Figure 3-1: The proposed system registration and authentication processes.....	39
Figure 3-2: Verification processes.....	41
Figure 3-3: A simplified Entity Relationship (ER) diagram of the relational database used for the proposed system.....	43
Figure 3-4: Structure of the user's table in the proposed system's database.....	43
Figure 3-5: Structure of the mobile device information table in the proposed system's database.....	44
Figure 3-6: Structure of the collected data by mobile device table in the proposed system's database.....	44
Figure 3-7: Structure of the overlapping access points table in the proposed system's database.....	45
Figure 3-8: Login machine application.....	47
Figure 3-9: Mobile device application.....	48
Figure 3-10: Activation of an account.	48
Figure 3-11: Communication system overview.....	49
Figure 3-12: Overview of proposed system.....	52
Figure 3-13: An example of a list of overlapping access points.....	54

Figure 3-14: An example of the distances between a user’s devices and four access points which are identified by red boxes.	56
Figure 3-15: A diagnostics screen for testing purposes.	57
Figure 4-1: Process of an asymmetric key encryption algorithm	67
Figure 4-2: One-time login page.....	75
Figure 4-3: OTP	75
Figure 4-4: Session termination.....	78
Figure 5-1: Netgear R6400 Wi-Fi router [67]	81
Figure 5-2: Illustration of the access point setup.....	84
Figure 5-3: Data collected before it is run through RapidMiner.	88
Figure 5-4: Euclidean distance [70].	92
Figure 5-5: Manhattan distance [70].....	92
Figure 5-6: Total time of logging in (in seconds) for the proposed system.....	99
Figure 5-7: Total time of logging in (in seconds) for different types of 2FA systems [74]	100
Figure 5-8: Result of testing the proposed system’s robustness through implementing the verification processes.....	101
Figure 5-9: Rejection request.....	104
Figure 5-10: Battery consumption.....	105
Figure 5-11: Battery consumption.....	106

ACKNOWLEDGMENTS

First and foremost, all praises and thanks are due to Allah the Almighty. For without His graces and blessings, none of my goals would be accomplished. Thank you Allah (SWT) for guiding me during my academic years and helping me complete this dissertation successfully.

I owe my deep gratitude to my advisor Dr. Jean Gourd for all his help and support during my academic journey. I am also thankful and fortunate enough to receive encouragement, support, and guidance from Dr. Manki Min, Dr. Miguel Gates, Dr. Jinyuan Chen, and Dr. Ibrahim Abdoulahi for being part of my committee. Also, tremendous guidance and assistance came from Prof. Galen Turner and Prof. Weizhong Dai. I am extremely privileged to have received such support during the duration of this project.

I would like to express my deepest gratitude to my lovely wife Dr. Thamraa Alshayeb and kids for being there for me. Thank you all for your patience and understanding during the countless hours of studying. Words are not enough to express my appreciation.

My heart felt regard goes to my siblings (Saeed, Fatemeh, Reem, and Mohammed) for their love and moral support.

I offer my sincere thanks and appreciation to my uncles Prof. Ali Darrat, Prof. Khaled Al-Agha, and Engr. Nasri El-Awadi; and to my aunts Om Faraj and Om Zakaria.

I pray to Allah (SWT) to bless them with lots of happiness, good health, and a long, healthy life. Also, my special thanks are extended to my brothers Dr. Aadel Darrat, Dr. Mohamad Darrat, and the future doctor Zakaria El-Awadi for their support and encouragement.

I would like to thank the local Muslim community for their friendship and support. You have made Ruston my home away from home. Also, I am very grateful to my brothers in the NCF Committee; their friendship has been a source of spiritual growth and has made my stay in Ruston more enjoyable.

My special regards to all my previous teachers in life. You have made it possible for me to see this day. Your kindness and patience allowed me to reach a stage where I could write this dissertation.

Finally, I am greatly indebted to my friends and to my colleague for the invaluable support and encouragement. To everyone who supported me, gave me advice, or had any contribution to accomplishing this work, thank you from the bottom of my heart.

CHAPTER 1

INTRODUCTION

Smart devices (phones, laptops, tablets, etc.) are electronic devices that are able to connect, share, and interact with each other via a variety of wireless protocols (e.g., Bluetooth, Zigbee, near-field communication (NFC), Wi-Fi, LiFi, 3G, etc.). Furthermore, they can operate, to some degree, autonomously. The advantage of these smart devices lies in their convenience and ubiquity. As of 2018, according to the Pew Research Center, 96% of Americans own a cellphone, and 81% of them use smartphones as their primary means of online access; this practice is especially common among young adults [1]. In addition, nearly three quarters of American adults own traditional computer systems, while around half of Americans own tablets [1]. These smart devices can receive signals sent using different radio frequency technologies and from different networks within range. As the devices move between networks (e.g., wireless access points, cellphone towers, etc.), they receive broadcast messages. If the broadcast messages originate from a Wi-Fi (IEEE 802.11) network, they include several useful pieces of information that are relevant to this work, and which are discussed in the technical background section below. Specifically, this information can be utilized in a variety of ways, such as to establish a connection, share information, locate devices, and, most appropriately, identify users.

Electronic authentication involves confirming a user's identity based on a claim made by the user to an information system [2]. The process contains two phases:

(1) the identification phase, and (2) the verification phase. In the identification phase, the system links a user to an identity through information provided by the user (e.g., a username). In the verification phase, the user's identity is validated using another piece of authentication information (e.g., a password) [3]. In order to establish and trust the identity of a user, multiple factors should be used simultaneously during the authentication step. As described above, the information provided by Wi-Fi access points can be used in many ways. One use involves an additional layer of user authentication that effectively acts as a second factor. In the following section, we will study the different types of authentication credentials that can be used to verify the identity of users.

1.1 Authentication Credentials

As cyberattacks have evolved, a diverse set of authentication methods has been developed to protect against them. The most common authentication methods fall into two categories: single-factor authentication (SFA) and two-factor authentication (2FA) (also known as dual-factor authentication (DFA)). A factor is considered to be one of the following characteristics: some place where a user *is*, something a user *possesses*, something a user *is*, something a user *knows*, or something a user *does*. These five elements are called authentication credentials.

1.1.1 Something a User Knows

When a user logs into an information system, the first task of authentication involves identification. This can be done by providing a username, for example. In order to authenticate the client, the client must prove that he/she is the actual owner of an

existing profile. This step could be done by providing something no one but that user knows, such as a password.

Currently, passwords are the most common means of authentication. Nevertheless, a password alone is often considered a weak form of data protection [4]. Moreover, there are numerous types of attacks that target “something that a user *knows*,” such as brute force and dictionary-based attacks. In response to these attacks, users should take action to strengthen passwords. For example, to achieve higher security for user information, passwords should be unique and should contain upper- and lower-case letters, numbers, and special characters.

1.1.2 Attacks on Passwords

A user can follow the rules for making a password strong, but that does not necessarily mean that attackers will be thwarted. Listed below are some attacks that attackers have used to gain information related to user passwords.

1.1.2.1 *Social engineering*

This type of attack exploits weaknesses in human interaction and manipulates people to break through security barriers, gain access to systems, locations, or networks, or for financial gain. Phishing, shoulder surfing, and dumpster diving are some types of social engineering attacks that currently exist. Phishing is a method of obtaining personal information—such as a password—by disguising oneself as a truthful entity. Shoulder surfing is a technique for gaining a victim’s personal information by spying on a user or looking over a user’s shoulder. Dumpster diving is a scheme in which an attacker attempts to retrieve information from a victim’s trash or belongings, which may then be used to launch an attack.

1.1.2.2 *Capturing a password*

There are several ways to capture passwords from a network or a user device. On a network, for example, passwords can be obtained using man-in-the-middle or relay attacks. In these types of attacks, a malicious user inserts himself between two parties and obtains credential information such as a password. Furthermore, a man-in-the-middle attack involves an attacker compromising and potentially altering the communication between two parties. Relay attacks involve sniffing for valid data that has been sent by an authorized user, and subsequently relaying it to a destination host. This type of attack happens when a malicious user detects a network data packet, and then intercepts it and relays it as if it were his own. A keylogger attack is a type of attack that is implemented on the user's end, where a password is captured using an application that listens to keystrokes on a computing system.

The use of the aforementioned attacks is limited, because the attacker requires access to a user's computer, or the attacker must observe a user entering a password. In both cases, the attacker must be physically present and in close proximity to the unsuspecting user. However, many types of attacks require little to no communication with the targeted system. These types of attacks are called offline password recovery attacks. Examples include brute force attacks, dictionary attacks, hybrid attacks, and the use of rainbow tables. These are discussed below.

1.1.2.3 *Brute force attack*

This attack involves the largely automated submission of every possible combination of letters, numbers, and special characters to attempt to match (and determine) a user's password.

1.1.2.4 *Dictionary attack*

This type of attack utilizes a dictionary (a list of candidate passwords) to discover a user's actual password. Obviously, the user's password must be in the dictionary for the attack to be successful. However, it is possible to combine words from the dictionary to form more complex passphrases that the user may have used during authentication. This "hybrid attack" is discussed below.

1.1.2.5 *Hybrid attack*

This type of attack is a blend of both brute force and dictionary-based attacks. Characters are combined using a dictionary. Moreover, concatenations and/or combinations of words in a dictionary can also be implemented to attempt to determine a user's password.

1.1.2.6 *Rainbow tables*

These "tables" are simply pre-computed hashes of potential user passwords. Most secure database systems do not store user passwords directly. Instead, they store the one-way hashes of user passwords. Because generating the hash of a password can be time consuming, it is often intractable to brute force or use a dictionary-based attack on such hashes. Rainbow tables can instead be used to make a direct comparison of the hashes stored in the databases of systems that are used to authenticate users. Although creating rainbow tables may take a significant amount of computational time, it only needs to be done once. There are many datasets of rainbow tables available online for the security community (and also for potential attackers).

Each method of attack discussed above has unique characteristics; however, they all have the same end goal: to determine a user's password. When sensitive user

information is protected by a password that can be compromised, a better way to defend against an attacker is needed.

As cyber-attacks have evolved, passwords have become insufficient to protect user information. Without the appropriate precautions, which can be two factors of authentication, for example, an attacker need only determine a user's password to gain access to a system. Protecting data using only a password, considering the above methods of attack, is almost, in some cases, deemed unwise. However, there are methods for decreasing the vulnerability of a system by utilizing an extra layer of security; for example, by including another authentication factor: something that a user *possesses*.

1.1.3 Something a User Possesses

This category of authentication is based on items that a user physically *possesses*. These items are used along with passwords. Due to the usage of two types of authentication, what a user *knows* (a password) and what the user *has* (e.g., a mobile device that is typically carried), this type of authentication is usually referred to as two-factor authentication (2FA). However, if a user were to use only one of the factors, it would be considered SFA (single-factor authentication). At present, the items that are commonly used to authenticate users are tokens, smart cards, and mobile devices.

1.1.3.1 Tokens

A token is a small electronic device with a light-emitting diode (LED) display screen. A token can be used instead of a password (what a user *knows*), and is a form of authentication that depends on what a user *has*. The method in which a token can be used is either to receive a one-time password (OTP) or to generate a OTP. There are two types of OTP: (1) a time-based OTP (TOTP); and (2) a hash-based message authentication

code (HMAC)-based OTP (HOTP). A TOTP is a temporary passcode that changes after a set period of time. Usually, a TOTP is generated and displayed on the token's LED screen once for a limited time. Because the token may not have a direct connection to the authentication entity, a shared algorithm between the token and the corresponding authentication entity would use the current time of day as one of its authentication factors. When a user logs in with a username and password, the code is then displayed on the token. After the user enters the token that is displayed, the authentication entity receives it and verifies that they are identical. The code must be used within the token's predefined time limit. After the token's time limit is up, it can no longer be authenticated by the authentication entity, and the user is not allowed to access another session with that token. If the user is already logged in with the expired token, the user is then logged out.

An HOTP is a OTP that changes when a specific event occurs. For example, when a user enters a personal identification number (PIN), this would then trigger the token to generate a new passcode.

Tokens have some benefits over passwords. First, a password is static; it does not change unless the user changes it (or they are forced to do so), which gives an attacker a chance to recover it and subsequently use it. Alternatively, a token frequently generates a dynamic password that is valid for only one operation within a specific time, which creates a barrier to unauthorized access.

1.1.3.2 *Smart cards*

Smart cards are plastic cards that contain a microprocessor chip that holds data related to a user. Smart cards are sometimes known as chip cards or integrated circuit

cards (ICCs). These smart cards can be used as personal identification, for authentication, and even for data storage. Smart cards can be implemented as contact cards, which requires physical contact, or contactless cards, where the information in the chip is transferred electronically.

1.1.3.3 *Mobile devices*

Today, mobile devices have overtaken tokens and smart cards. A code can be retrieved through an application on a mobile device, or as a text message that is sent to a user's cellphone. In addition, through a mobile device, a user can ask to receive an HOTP passcode.

To add an extra layer of security, we can use not only something that a user *possesses*, but also something that a user *is*.

1.1.4 Something a User Is

This type of authentication depends on the features and characteristics of a person. These features and characteristics can either be physical or non-physical. This form of authentication is known as biometric authentication. Physical biometric authentication is referred to as standard biometrics, while non-physical biometric authentication is referred to as cognitive biometrics.

1.1.4.1 *Standard biometric authentication*

Standard biometric authentication is a security process that relies on the distinctive physical characteristics of a user (what a user *is*) to verify that a user is who they claim to be. Standard biometric authentication can use fingerprints, palm profiles (i.e., veins in the palm), face recognition, palm prints, retina recognition, hand geometry, or iris recognition. Fingerprint and face recognition have become the most common types

of biometric authentication used today. However, fingerprints are more often used than face recognition [5]. There are two different types of scanners for fingerprints: a static fingerprint scanner and a dynamic fingerprint scanner. With a static fingerprint scanner, a user must place the whole thumb or finger on an oval scanner window. A dynamic fingerprint scanner requires a user to place a part of a thumb or finger on a small scanner window.

1.1.4.2 *Cognitive biometric authentication*

Cognitive biometric authentication is a method that recognizes individuals based on their thoughts, perceptions, and observation processes using responses from nerve tissue. Nerve tissue is a major component of the nervous system, which consists of neurons [6]. The responses from a user's nerve tissue can be obtained using many techniques; for example, by using an electroencephalogram (EEG), an electrocardiogram (ECG), or from an electrodermal response (EDR). These techniques are a way of recording brain activity that is produced during a practice activity, which can be either a mental or physical activity.

In general, biometric authentication has its advantages. The first is better quality and security: using a unique feature of the human body for authentication creates an obstacle for an attacker that cannot be predicted: a user must be present to gain access. Second, a user's biometric identity cannot be lost: because this category of authentication depends on a part of the human body, it is typically difficult to lose it or not have it. The third advantage is convenience: it eliminates many of the difficulties of the identification methods that are associated with what a user *knows* or what a user *possesses*, making authentication truly convenient. Nevertheless, biometric authentication has some

disadvantages as well. For example, it is costly: a biometric authentication system typically costs a substantial amount of money to set up [7]. These costs could be due to both software and hardware requirements, and the system may even incur additional costs to maintain in the future. Also, there is some technical complexity: some organizations may not be in a position to have a high-level technician maintain and perform day-to-day front-end and back-end operations. The major disadvantage of a biometric authentication technique, however, is that once a system with a biometric signature is compromised, a user cannot “reset” or change his or her body. Because of this, it is advantageous to consider relying on something else to secure data; for example, the behavior of users (i.e., something that a user *does*).

1.1.5 Something a User Does

This is another type of authentication factor that is based on identifying and measuring patterns based on a person’s behavior while performing certain activities. This kind of authentication is known as behavioral biometrics. Keystroke dynamics and voice recognition are two examples of behavioral biometrics.

1.1.5.1 Keystroke dynamics

Keystroke dynamics uses recognition of the patterns of rhythm and timing created when an individual types on a keyboard. This method uses many measurements to study a person’s typing behavior, such as a dwell time: the duration of time that a key is pressed; flight time: the time duration in between releasing a key and pressing the next key; and common error: the errors that occur when a user types in a username or password, for example. The advantages of keystroke dynamics are its affordability (it does not require additional hardware parts or even a user interface), convenience (a user

does not need to take an extra step to be authenticated), and reliability (because this method is based on a user's typing behavior, it can never be lost or deleted). Keystroke dynamics also has some disadvantages. For example, it generally suffers from low accuracy (a user's typing pattern may change due to injury, exhaustion, or distraction), and it has low permanence (a user's typing pattern may gradually change over time because they may get accustomed to typing a password, adapt to an input device, or increase their typing proficiency).

1.1.5.2 *Voice recognition*

The voice recognition technique analyzes a person's voice to identify the unique characteristics to use when verifying their identity. Several characteristics, such as age, the shape and size of the mouth, the shape of the head, and the movement of one's jaw create a unique voice for every person. Like other types of factor authentication, voice recognition has advantages and disadvantages. Some of the advantages of voice recognition include convenience (it has high social acceptability), cost-effectiveness (there is no extra hardware required), and it is widely accessible (most devices are equipped with a microphone, for example). However, the disadvantages are that it can easily be spoofed (the voice is prone to spoofing, where an attacker can record a user's voice and use the recording for authentication), there is a time cost (in order to verify a user's identity, the system may require some time), and accents may affect the result (voice recognition may not be able to recognize a user's voice who is not a native speaker of the system's language, for example).

Lastly, some users may not feel comfortable speaking aloud in public to verify their identity. Because of this, some users tend to use another authentication factor to prove their identity; for example, the user's location (i.e., somewhere a user *is*).

1.1.6 Somewhere a User Is

This type of authentication is based on a user's location, and is known as geolocation. Although location may not always be considered a unique identifier of a user, it can be utilized to indicate if a remote adversary is attempting to perform some sort of malicious activity from a location that is not the typical location of the user.

For example, suppose that a user usually accesses her bank account from home when she gets home from work. In this case, the bank's remote authentication entity can use this information to establish a geolocation pattern based on the Internet Protocol (IP) address of her computer. If remote access is attempted from another country, for example, this may indicate a malicious action. Geolocation authentication is used by most banks, for example, to block wire transfers from overseas unless the user has informed the bank to approve any transfers from another geolocation ahead of time. This method of authentication is used not only by banks; many websites will not allow a user to gain access if the user's location is not in the customary location.

In summary, there are five elements of authentication that, together, comprise authentication credentials. These five elements include *somewhere* that a user *is*, *something* that a user *possesses*, *something* that a user *is*, *something* that a user *knows*, and *something* that a user *does*. A significant goal of this work is to introduce and utilize a sixth element: *something* that is in a user's *environment*. One of the goals of this work is to add a second layer of authentication that utilizes the current IEEE 802.11

infrastructure to prove a user's identity without adding any additional burden on users of the system. Because there is no additional burden on the users, something that we believe is novel in our work, we call our system a zero-effort two-factor authentication (0E2FA) system. In the following section, the IEEE 802.11 infrastructure will be discussed, including how we plan to utilize it to enable our proposed method.

1.2 Technical Background

IEEE 802.11 refers to the most popular set of standards for wireless local area networks (WLAN) [8]. More commonly known as Wi-Fi, the Institute of Electrical and Electronics Engineers (IEEE) adopted it as the first WLAN standard in 1997 [9]. The Wi-Fi protocol links two or more devices using radio frequency wireless communication, and usually requires shared connections over access points [10]. The hardware that allows devices to connect with an access point is commonly called a Wi-Fi card, Wi-Fi device, Wi-Fi interface, or a station. Over its history, there have been several different revisions of the Wi-Fi standard (e.g., 802.11a, 802.11b, 802.11g, 802.11n, and recently, 802.11ac and 802.11ad). These specifications differ in terms of speed, performance, and range [11].

WLAN has the ability to operate over bands of 2.4 GHz, 3.6 GHz, or five GHz. [12]. A band is the frequency range of the electromagnetic spectrum that is utilized to send a signal, and it is strictly regulated for specific use by the Federal Communications Commission (FCC) [7]. The data transmission speed over Wi-Fi also varies based on the protocol revision, ranging from six Mbps in IEEE802.11a, all the way to 600 Mbps in 802.11n. Wi-Fi bands are further divided into channels that allow various nearby devices to share the same frequency band without interference [7]. The 2.4 GHz band is more

likely to be subject to interference, however, as it is a popular “sweet spot” in terms of range and bandwidth (features that are inversely proportional). It is most commonly used by electronics applications such as Bluetooth, cordless phones, and wireless peripheral dongles, while also being emitted by some appliances such as microwave ovens.

Unfortunately, this band allows for very few channels to be simultaneously usable without interference. Conversely, the five GHz band has 19 available channels and greater bandwidth at the cost of penetration potential and range. Devices using this band must have the ability to examine the load on each channel and choose the best one for minimizing interference.

As mentioned in the previous section, this research aims to use IEEE802.11 access point broadcast information to provide an additional means by which to authenticate users. Typically, a broadcast message comes in the form of a beacon frame, a type of management frame that is transmitted periodically by Wi-Fi access points for the purpose of establishing connections [13]. The beacon frame announces the existence of a network, allowing devices to find and identify the network, and subsequently to request access. It contains the information about the network needed to make the request [14]. The time between beacon frame broadcasts differs from network to network, but the broadcasts come at regular intervals [13]. The range within which a device may receive a beacon frame is defined as the basic service area (BSA). Figure 1-1 shows the structure of a beacon frame in IEEE 802.11 that consists of three sections: (1) the medium access control (MAC) header, (2) the frame body, and (3) the frame check sequence (FCS) [15].

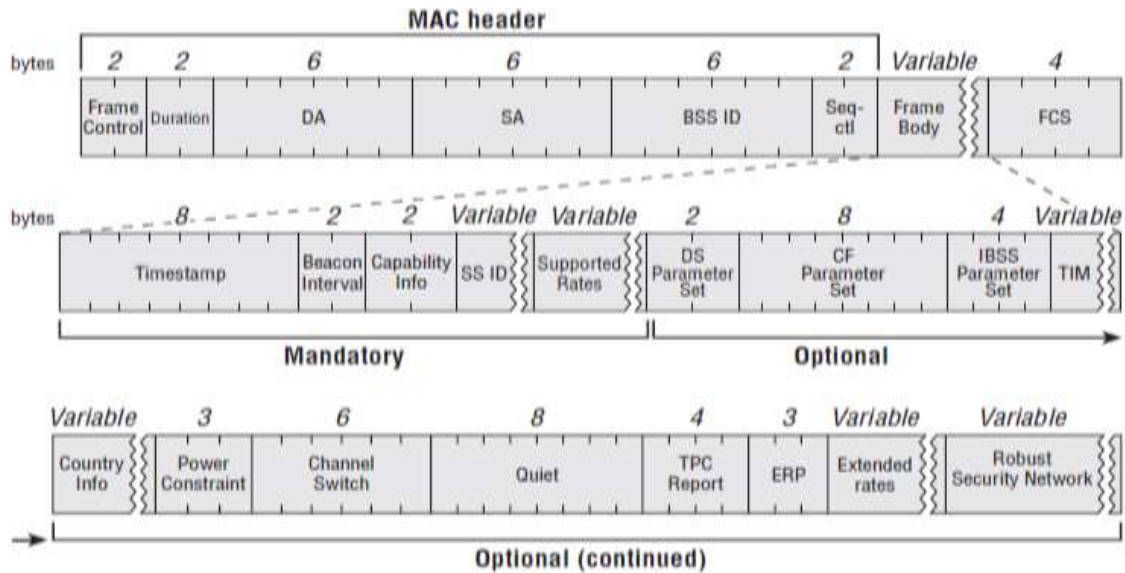


Figure 1-1: 802.11 Beacon frame [16]

1. **MAC header:** This header is comprised of multiple subsections, such as frame control, duration, address, and sequence control information. The header will also include quality of service (QoS is any technology that controls data traffic on a network [17]) data frames, and QoS control information, which is a 16-bit field [18]. In the proposed system, the MAC address (also known as the BSSID) will be used to verify that a user's devices are co-located. More details will be provided further in this text.
2. **Frame body:** The frame body is a field of variable length that contains information particular to the frame body type and subtype [17]. It is used to carry several pieces of information pertaining to the network that are needed by stations requesting access. The frame body consists of the following fields:
 - a. **Timestamp:** The timestamp provides the timing synchronization function (TSF) that represents the frame's creation time [17]. A TSF keeps the timers

of all stations in the same BSA synchronized, and all stations should maintain a local TSF timer [17].

- b. Beacon interval: This represents the number of time units between beacon transmissions, which can be used to determine the next time to check for incoming traffic. This is useful, for example, when devices go into power-saving mode [19]. This information allows an access point to switch its Wi-Fi radio unit on and off several times per second to save power (usually in the form of a battery). For most access points, the beacon interval has a default value of 100 ms; however, it can be adjusted in most cases [19]. Access points must schedule their beacon transmission at the selected beacon interval; however, the transmission may suffer some delays while attempting to avoid a collision with the beacon frames broadcast by other devices [19].
- c. Capability information: This information contains a number of subfields used to indicate connection requests or to advertise optional capabilities for those devices wishing to connect to a wireless LAN [19].
- d. Service set identifier (SSID): The SSID is a unique identifier that acts as the name of a particular WLAN. It is utilized by clients to seek out a preferred network in order to establish a connection [20]. The SSID is sometimes known as the extended SSID (ESSID) [21]. For an additional layer of security through obfuscation, the user may choose to have the access point leave the SSID out of the beacon frame [20], although this does not necessarily prevent attackers from gathering information about the WLAN. More details will be

provided further in Chapter 3 Section 3.3.5 that explains how the proposed system utilizes the SSID to verify that a user's devices are co-located.

- e. Supported rates: Each broadcasting access point supports connection speeds [19]. Stations can use this information to decide which access point is most suitable for a connection [19].
 - f. Parameter sets: This field specifies the signaling methods to be used, such as frequency hopping spread spectrum, direct sequence spread spectrum, etc. A beacon would include the appropriate signaling methods to adapt to the channel's condition and to achieve efficient data transmission [22].
 - g. Traffic indication map (TIM): Every beacon frame contains a list known as a TIM. A TIM is a list of all the stations that have buffered frames awaiting delivery at an access point [22].
3. FCS: This sequence is a 32-bit field that contains a cyclic redundancy check (CRC). The CRC is used to validate the integrity of received frames [18].

In this study, we aim to utilize the Wi-Fi broadcast messages and specifically selected information contained in the beacon frame to implement a 2FA system that requires no effort on the part of the user. Broadly, the latter characteristic is one important distinction that makes this work different and unique from others. For example, a user will not be required to receive and subsequently enter a code, or to confirm a login attempt using a mobile device. The beacon frame specifics are used to determine if two devices are in the same "environment." Specifically, the SSID and BSSID are used alongside the value of the received signal strength indicator (RSSI) as unique identifiers.

The SSID is used to name a wireless network and can be up to 32 characters long. Both the client's device (i.e., Wi-Fi card) and access point (e.g., router) must share the SSID. The SSID is broadcasted in an area to present the existence of the wireless network. Before linking with a specific wireless network, a device known as wireless network interface card (Wi-Fi NIC) must pair with the SSID of the access point. Figure 1-2 shows an example of an SSID list that was scanned in a specific location (more details will be provided in Chapter 3 Section 3.3.5).

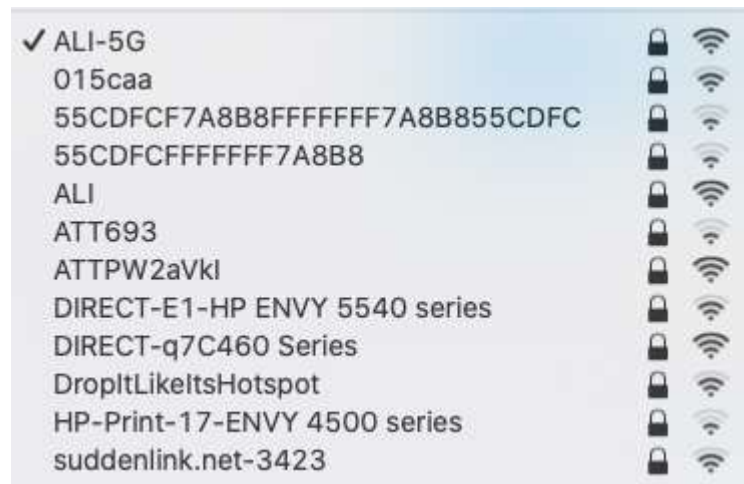


Figure 1-2: A list of access points showing the SSID and signal strength of each

The BSSID is a hardware identification number that uniquely identifies an access point. The format of the BSSID in a broadcasted message can be represented as a MAC address using any of the following formats [23]:

1. 00:00:00:00:00:00,
2. 00-00-00-00-00-00, or
3. 00.00.00.00.00.00.

The first six-digits of the MAC address represents the manufacturer and is called the Organizational Unique Identifier (OUI). Below are some examples of well-known manufacturers [24]:

1. CC:46:D6 - Cisco
2. 3C:5A:B4 - Google, Inc.
3. 00:9A:CD - Huawei Technologies Co., LTD

The remaining six digits are a combination that is uniquely allocated to each Wi-Fi NIC [23]. Figure 1-3 shows an example of a BSSID list.

Network Name	BSSID
ALI-5G	1c:49:7b:e0:61:38
ATT693	ac:5d:10:4c:69:2a
ATTPW2aVkl	88:ef:16:a5:58:d0
DIRECT-E1-HP ENVY 5540 series	18:60:24:30:03:e3
DIRECT-q7C460 Series	32:cd:a7:1c:ae:1e
DropitLikeltsHotspot	8c:3b:ad:e6:94:92
suddenlink.net-3423	2c:99:24:a0:34:21

Figure 1-3: List of SSIDs (Network Name column) with their respective BSSIDs

As any signal travels, it decays, or decreases in strength. This decay is relative to the transmitter and receiver, and is represented by the RSSI (Received Signal Strength Indicator). The RSSI that is measured at the receiver's antenna is a value that can be used to calculate the distance from the transmitter to the receiver. Thus, the RSSIs (expressed in dBm) perceived at the receiving station are consistent with the station location with reference to the transmitting access points. Figure 1-4 shows the beacon information and corresponding RSSI values received by a MacBook Pro that is monitoring beacon frames from several access points. This information was collected using the built-in wireless diagnostics tool within MacOS Catalina. An RSSI value is always negative, and it

translates to a very small but positive number that corresponds to power on a logarithmic scale [25]. A negative dBm means a negative exponent is applied in power calculations (e.g., -10 dBm equates to 0.1 mW, -20 dBm equates to 0.01 mW, etc.) and represents a loss of signal strength between the sender and the receiver [25].

Summary	Network Name	BSSID	Security	Protocol	RSSI	Noise	Channel	
Total	12	015caa	bc:2e:48:5c:0f:5e	WPA/WPA2 Personal	802.11ac	-82	-91	1
2.4 GHz Count	11	55C0FCFFFFFF7A8BB	fc:a9:34:1d:09:90	WPA2 Personal	802.11b/g/n	-86	-91	1
5 GHz Count	1	9m1e	3c:37:96:4b:cfa0	WPA2 Personal	802.11b/g/n	-88	-89	4
Current Channel Count	1	9m1e-Quest	3c:37:96:4b:cfa0	WPA2 Personal	802.11b/g/n	-89	-89	4
Best 2.4 GHz	1	ALI-99	1c:49:7b:e0:01:...	WPA2 Personal	802.11ac	-67	-91	148
Best 5 GHz	101	ATT693	ac:5d:10:4c:89:2a	WPA/WPA2 Personal	802.11b/g/n	-75	-91	6
		DIRECT-51-HP ENVY 6540 series	18:60:24:30:03:e3	WPA2 Personal	802.11b/g/n	-82	-91	1
		HP-Print-85-Officejet Pro 8800	2c:59:ad:af:99:85	Open	802.11b/g	-85	-89	6
		NETGEAR66	b5:71:b8:77:66:33	WPA2 Personal	802.11b/g/n	-88	-91	11
		suddenlink.net-3423	2c:99:24:a0:34:21	WPA/WPA2 Personal	802.11b/g/n	-74	-91	11
		suddenlink.net-634A	c0:c5:22:fd:63:48	WPA/WPA2 Personal	802.11b/g/n	-87	-89	11
		suddenlink.net-BD5D	1c:ab:c0:5c:8d:52	WPA2 Personal	802.11b/g/n	-86	-90	1

Figure 1-4: Beacon frame information for several access points

As mentioned earlier, the SSID and BSSID can be used as an indicator that a user's devices are in the same physical location. The RSSI value is used to calculate the distance between a device and an access point (more details on this are provided in Chapter 3 Section 3.4.2). Both an SSID and BSSID are needed because if two access points have the same SSID, then the BSSID can be utilized to differentiate between them. This is because the BSSID is unique as explained previously in this text, and it is impossible for two access points to have the same BSSID (as explained previously in this section). Amongst the beacon frame's characteristics, the SSID and BSSID were chosen to determine the location of a user's devices because the SSID and BSSID are both unique identifiers of an access point. These two characteristics are also used when announcing the presence of the access point in an area which can then be used for the purpose of communication [23]. Other than the SSID and BSSID, the other beacon frame

characteristics do not help to improve the proposed system's accuracy (more details about this are provided in Chapter 5 Section 5.2).

1.3 Contribution

Using the current IEEE 802.11 infrastructure, the proposed system implements a second layer of authentication and places no burden on users. Moreover, in this dissertation a zero-effort two-factor authentication system based on something that is in the user's environment (ambient access points) is proposed. In this research, data from the broadcast messages are utilized to implement the second authentication factor by determining whether two devices are in the same physical location, in a way that requires zero interaction from the user.

The proposed solution aims to add an extra layer of authentication by finding something unique that is in a user's environment; specifically, by utilizing information at ambient access points (a Wi-Fi footprint). This system depends on something that a user knows, something that a user owns, and something that is in the user's environment – arguably the most significant contribution of this work. Primarily due to the proposed system removing the “weak link” from the second authentication factor, the proposed system facilitates the use of 2FA in more systems.

1.4 Guide to Dissertation and Conclusion

In this chapter, authentication credentials were discussed. Furthermore, some examples of authentication credentials were introduced. Also, the relevant technical background was discussed. Chapter 2 discusses existing research in the field of 2FA, the weaknesses of these ideas and how our proposed system can address them where

relevant. Chapter 3 introduces the proposed system and discusses some of the challenges encountered during its development and address the shortcomings that have been mentioned in the various related works in Chapter 2. Chapter 4 evaluates different aspects of the proposed system. Moreover, the different factors and parameters that play a role in the proposed system's performance and security are analyzed. Chapter 5 outlines the design specifics of the proposed system and the challenges faced during the design process. Chapter 1 presents numerous experiments that were utilized to test the proposed system's design in various aspects. Chapter 6 summarizes the proposed system and discusses possible future directions.

1.5 Peer Review

Portions of the proposed system have been accepted and/or published at various conferences and in various journals or are currently being reviewed. Specifically, the results of experiment one (discussed in Chapter 5) were presented in 2020 3rd International Conference on Applications & Information Security (ICCAIS 2020) Riyadh, Saudi Arabia, March 2020 [26]. Also, the same results were developed and published for the IEEE Access journal [27]. In addition, a miniature version of the proposed system was submitted to two conferences and were accepted (i.e., IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC 2020), Honolulu, Hawaii, USA, Aug 2020 and The 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud 2020), Aug 2020, New York City, NY, USA [28,29]).

CHAPTER 2

BACKGROUND

In this chapter, we discuss existing research in the field of 2FA to differentiate these ideas from our proposed system. Furthermore, we discuss the weaknesses of these ideas and how our proposed system can address them where relevant.

2.1 Related Works

In this section, the relevant existing ideas in the field of 2FA are presented, ordered according to the technology used at the authentication phase. The systems in Section 2.1.1 depend on a OTP. Section 2.1.2 presents authentication methods that depend on a phone call. Section 2.1.3 presents 2FA systems that depend on radio frequency ID (RFID) technology, Section 2.1.4 presents 2FA systems that depend on Quick Response (QR) technology, Section 2.1.5 presents 2FA systems that depend on NFC technology, and Section 2.1.6 presents 2FA systems that depend on ambient sound authentication.

2.1.1 Authentication Systems that Depend on a OTP

An OTP is a password that is sent to a user's phone, an email, or that is displayed to a user for the purpose of adding an extra layer of security. The OTP remains valid for a specific period of time [30]. This idea has been widely used in 2FA authentication systems to address vulnerabilities associated with traditional password-based authentication [31]. The following discussion includes some prominent ideas in this field.

In 2011, Yu and Brune proposed a 2FA scheme [32] entitled “No security by obscurity - why two-factor authentication should be based on an open design.” In the registration phase, the user selects from a list of OTPs that has been generated by an authentication entity. On the client side, software is installed to decrypt the OTP list to obtain the next valid OTP by submitting a valid username and password. In the login phase, a user submits the credential information. Subsequently, a sequence number is presented by the system that indicates which OTP in the list should be used. In the client software, the user enters the OTP along with a password that is designed to decrypt the OTP list in the registration phase. The software decrypts the OTP and displays it to the user. The user inserts the corresponding OTP, along with a user ID, within a predefined period of time in order to be authenticated. The authentication entity then checks the information and renders an authentication decision. The major security threat with this method is that there is a possible disclosure of the encrypted OTP list from the user’s side. For example, the user’s cellphone could be stolen or lost, and the user does not realize that this has happened. In this case, the OTP list could be copied or captured and used in an offline password guessing attack (an offline technique that an attacker attempts in order to discover a password by systematically trying every possible combination of numbers, letters, and symbols) which may lead to the attacker gaining access. In addition, although the system uses a strong encryption method (e.g., Twofish), there is still a weakness in which an attacker could decrypt the OTP during its period of validity by using the offline password guessing attack to break the encrypted password, and use it to gain access to the authentication entity [32].

In 2018, Al-Sahwan et al. presented an authentication protocol that depends on both the PassText mechanism and OTP techniques [33]. The PassText mechanism is essentially a meaningful text file that is presented to a user and replaces a password. During the registration phase, the user receives a PassText that must be modified by adding, deleting, or changing text, and that is stored on a remote authentication entity after it is encrypted with bcrypt and sha256 [33]. The user logs in to the system with a username only, and then the user is prompted with the original text file. The user then has to apply the same exact modification to the original file. After the user performs the modification, the authentication entity compares the hash of the submitted PassText with the stored PassText hash. A OTP is subsequently sent to the user's email or mobile device number chosen during the registration phase. In order to gain access, the user must enter the OTP correctly before the session ends or user access will not be granted. A weakness of this system is that a user can forget the PassText modifications. In this case, the user will need to reset it. According to Al-Sahwan et al, this can be done by sending a OTP to the user's cellphone number or email address. If the user enters the OTP correctly, s/he will be able to enter modifications on the original file. In this case, an attacker may request to reset modifications where the OTP could be easily obtained by using a phishing attack, for example. Moreover, remembering the modifications and waiting for the OTP to be sent is a significant inconvenience for users.

2.1.2 Authentication Systems that Depend on a Phone Call

A phone call is a connection between two parties over a telephone network. The first phone call was made on March 10, 1876, when Alexander Graham Bell called his assistant in the next room [34]. Today, making a call is as simple as dialing a number.

With a call, we can prove the identity of a user either with a voiceprint or a phone number. Because of this some authentication methods have used phone calls on authentication systems, as we discuss below.

In 2013, Fujii and Tsuruoka proposed a 2FA system called SV-2FA that used a one-time, disposable phone number along with an oath text sent via SMS to authenticate a user with voiceprint match [35]. During the registration phase, users must fill out a special form giving their ID, cellphone number, and an official signature or seal, and mail it to the organization managing the authentication entity. After the operator processes the user's information, the authentication entity sends a text message to the user's phone that includes an oath text and the one-time phone number, which can be called to proceed with the authentication process. Here is an example of the text message sent to a user: "We are going to begin the process written in the oath text below. If you consent to this, please call <phone_number> and read it within 45 seconds. OATH TEXT: 'I, <name>, register my voiceprint to <entity>. Today is <date>.'" [35]. For this process, the user calls the one-time phone number and reads aloud the oath text, the voiceprint of which is recorded along with a timestamp, an ID, and the originating phone number. The registration is then completed. The next time the user logs into the registered system, an SMS text message that contains a one-time return phone number as well as an oath text is sent to the user's cellphone number. Subsequently, the client must call the one-time phone number and speak the message back to the authentication entity. The authentication entity records the spoken oath text, along with the session number, for comparison with the user's stored voiceprint. If the recorded voiceprint matches the one stored on the authentication entity, the user is granted access; otherwise, the user's login

attempt will fail. One weakness of this method is that personally identifying information must be written on a form and physically mailed to the authentication entity's operator. During transit, this information could be exposed to postal workers or be stolen by a third party. Furthermore, the user is required to verbally read the oath message, which would likely be on an unencrypted line that could cause a user's voiceprint to be recorded and maliciously used by an attacker. In addition, there is the potential for a message to be delayed due to network congestion. After all, most consumers (99% of Americans, according to a recent Pew study [12]) own mobile phones capable of SMS; furthermore, in large cities, SMS messages often get delayed by a significant length of time [36]. Finally, this method has considerable requirements in terms of user interaction (writing, mailing, talking, etc.).

In 2015, Sodhi proposed a method that used a dropped call as a 2FA [37]. During the registration stage, the user provides a cellphone number to an authentication entity providing the requested service. When the user attempts to authenticate to the authentication entity, it asks the user for the registered phone number. If the entry matches what is stored in the authentication entity's database, the system selects an available phone number randomly from the database and sends it to the user's phone as a OTP. On the client side, the user is prompted to make a dropped phone call to the given phone number within a specific time. At that point, the authentication entity can receive a dropped call through the chosen number within a specific period of time—and subsequently makes an authentication decision. This scheme is subject to a denial-of-service (DoS) attack, where an attacker could keep a mobile phone ringing for an extended time, for example. In addition, a user's phone may be compromised, and it is

possible that a malicious entity could initiate a phone call from that number to an arbitrary phone number (i.e., to impersonate the user, an attacker could attempt to make a dropped phone call to the selected arbitrary number that the system has chosen).

Moreover, the telephony network can be spoofed, which means that an attacker could send an arbitrary phone number to the phones in the authentication entity's phone pool as the caller's identity to potentially gain knowledge of all of the phone numbers used by the server in the authentication system.

2.1.3 Authentication Systems that Depend on RFID Technology

RFID is a wireless communication technology that is used to identify uniquely tagged objects or people [38]. At present, RFID is used to authenticate users and prevent fraud or unauthorized access to an information system. We discuss relevant works in this area that use RFID to authenticate users.

In 2017, Mathew and Divya published an authentication system that uses RFID to gain access to a building, a room, or an office as a second factor of authentication [39]. To open a door, a user must place their RFID tag in contact with a reader; in turn, the reader retrieves the tag information. On the authentication entity side, if the information matches what is in the database, it generates a OTP and sends it to the client side. The user opens an application that has been installed previously in a mobile device to read the OTP. Once the user enters the received OTP correctly in the door's keypad, the user gains access. In the proposed idea, the authentication entity is trusted by users most of the time [40]. So, this system is subject to phishing attacks as the generated OTP would be easy to obtain. For example, an attacker can impersonate the authentication entity (i.e., the attacker can act as the authentication entity by sending a request to the user through

an email) and can request the user to authenticate himself. If the user does so, the attacker will subsequently receive the OTP and can use it to gain access [40]. The proposed idea requires hardware parts that must be installed in every door, which would cost money, time, and effort. To open a door, a user must insert an RFID tag, wait for a OTP to be sent, and subsequently enter it on the door's keypad. Because of this, the system is not convenient for most users (i.e., it requires user interaction – something that we wish to avoid in our proposed system). Moreover, the system requires power and may not always function properly, which could leave a door completely locked and inaccessible or potentially remaining in the open position.

In 2017, Tombeng and Laluyan published an authentication system that requires implanting a RFID tag into a human hand [41]. This security system works by replacing the mechanical lock system with RFID technology. In their work, it was implemented for access control of motorcycles [31]. A user first implants an RFID card in his/her body using a sterile implant kit. Then, the user places the implanted RFID tag close to an RFID reader that has been installed on a motorcycle. A microcontroller receives the information from the RFID reader, validates the RFID ID tag, and provides alerts in the form of LED lights and a sounding buzzer if the validation is successful. The microcontroller simultaneously instructs the electrical system to start the motorcycle engine. The major weaknesses of this system are privacy concerns and health issues. The privacy concerns are that every RFID chip contains a unique identifying number for each individual that could be linked to personal information. Moreover, installing or replacing the RFID hardware for implanting requires bodily intervention which could potentially be tricky.

An implanted RFID is also incompatible with strong magnetic medical equipment, which could prevent users from using medical equipment when necessary.

2.1.4 Authentication Systems that Depend on QR Technology

A QR code is a type of matrix barcode or two-dimensional barcode that contains information that is readable by smart devices [42]. In 1994, the QR code was created by Toyota, and was used to track inventory of vehicle parts [42]. Today, QR codes are used in many other fields such as commercial tracking, entertainment, in-store product labeling, and in applications that are designed for mobile device users [42]. In the following section, we will discuss two systems that use QR codes to authenticate users.

In 2016, Rodrigues, Chaudhari, et. al proposed a QR code-based scheme [43]. Their method uses QR technology to create a second authentication factor. In order to create a user profile and store it in the system, users must first visit a remote authentication entity (usually a website) and submit their full name, email address, password, and the unique international mobile equipment identity (IMEI) number associated with their mobile phone. Subsequently, the remote authentication entity generates a QR code using the IMEI information and a random four-digit code. Users then use a special application installed on their mobile phone to scan the QR code and input the received string into the remote authentication entity to verify the information and complete the registration phase. To later authenticate, users perform the same QR code-scan and string-input procedure. If the entered string from the scanned QR code matches the information at the authentication entity, users are successfully authenticated. Again, this method requires a certain amount of effort on the part of the user, both in the setup phase and, more significantly, during repeated authentication attempts. Finally,

there is also the problem of interception, as the QR code is displayed on the user's monitor during the entire process, and can be read from a distance. A user may be subject to an impersonation attack, in which an attacker can steal the QR code from the user's monitor during registration or authentication [36]. With the exposed QR code, the attacker could potentially obtain the user's IMEI with a QR code reader (which is now standard on most mobile devices), and then use it to impersonate the user via spoofing. This would be difficult to discover and rectify in a timely manner [45].

In 2016, a 2FA scheme, called 2FMA-NetBank, was proposed by Pratama and Prima to serve internet banking [46]. In this work a user enters a username and password during the login phase on the login web page of the proposed system. The remote authentication entity verifies the input credential information by comparing it with information previously stored in a database during the registration phase. If the username and password are correct, the authentication entity then generates a random value as a challenge using the authentication entity's private key. The remote authentication entity then combines this random value along with the user's IMEI to compute a "response" with the user's public key to encrypt it and store it in the system's database. Afterward, the authentication entity converts the digitally signed value and encrypted challenge to a QR code to display to the user on a web page. Using an application on the user's mobile device that was installed during the registration phase, the user scans the QR code to decrypt it using the authentication entity's public key. The digitally signed random value is verified with the user's private key. By using a software token that is installed on the user's mobile device, a combination of the random value and IMEI is computed to find the response value. Then, the user manually enters the response value on the banking

system web page. The remote authentication entity compares the entered response with the stored response in the database to either authorize or deny access. The weaknesses in the proposed system are that the QR code is displayed to the user on the internet web page. Consequently, it could be relatively easy to obtain from the screen or from the user's computer if the computer were compromised, for example. Because the software token is installed on the client side in order to decrypt the QR code during authentication, an attacker could gain that information if they were able to compromise the user's mobile device.

2.1.5 Authentication Systems that Depend on NFC Technology

NFC is a short-range wireless communication that allows smart devices to exchange data either by touching, tapping, or coming within a certain range of proximity [47]. In order for two smart devices using NFC technology to exchange information, both devices must be tuned to the same radio frequency. Moreover, both devices must be within operating range, which is typically between three to five cm [48]. Due to the short range of NFC, its use as a way to authenticate users is relevant. We review two such systems below.

In 2017, Hufstetler, Ramos, and Wang proposed a 2FA system that works by scanning an authorized NFC tag [49]. The method relies on pGina, an open-source credential provider replacement that allows developers to create plugins for alternate methods of authentication and to allow access to a machine that is running a Windows operating system. Moreover, an NFC tag must be attached to the user's system and be configured with pGina. During login, the user is asked to type a predefined passcode into a pGina login page. When the user is ready, the login page is submitted while

simultaneously holding an authorized NFC tag up to the reader. The entered information and the scanned NFC tag are sent to the plugin to check if they match the ones registered during setup. Depending on the outcome, pGina either denies or grants the user access to the remote system. Unfortunately, there are several weaknesses with this method – primarily due to the wireless nature of NFC [50]. Eavesdropping can occur during NFC communication if the attacker is close enough to the user’s device [51] [52]. NFC does not have any type of safeguard against the possibility of eavesdropping [52]. Also, because every NFC tag has a built-in unique ID, this ID can be spoofed, captured, and/or copied, as the ID is not encrypted. This security may not be sufficient to protect a user’s information [53] [54] [9]. Lastly, this system requires significant user interaction as well as a physical device (NFC tag) to keep track of. This is not convenient for users of the system (which is a weakness that we consider significant, and that we wish to address in our proposed method).

Another system that uses NFC, published in 2019, authenticates a user at an ATM machine by having the user swipe a mobile device in front of an NFC reader [55]. During the registration phase, the user installs a relevant application. Through the application, the user enters a username and a default PIN, and sends a registration request to an authentication entity. The remote authentication entity responds by sending a text message with a OTP to the user’s cellphone number. The user types the OTP into the application and submits it to the authentication entity for verification. After the OTP is validated, the user must select a new PIN and submit it through the application. In the meantime, the application collects information specific to the mobile device, such as device ID, make, model number, and operating system (OS) type, and sends this

information in the same message with the new PIN to be stored in the authentication entity's database. In the authentication phase, the user sends a username and password through the application to the authentication entity. After successfully validating the credential information, the authentication entity responds to the application with a OTP that is valid for only one transaction within a specific time; furthermore, the OTP is invisible to the user and internally stored in the application. When the user swipes his/her mobile device in front of the ATM NFC reader for authentication, the stored OTP is transferred from the application to the authentication entity. If the scanned OTP matches the one stored on the authentication entity, the transaction is successful; if not, the user is rejected. The weakness in this system is that the security in NFC channels is not sufficient enough to protect user information because all NFC channels are unencrypted and can be spoofed, captured, or copied [53] [54] [9].

2.1.6 Authentication Systems that Depend on Ambient Sound

Ambient sound refers to any sound that serves as background noise at a given location. Because every location has its own unique ambient sound, several researchers have utilized sound to authenticate users.

QuickAuth, a method proposed by Zhu, Yu, and Pei in 2016, depends on the ambient sound around the user during authentication [56]. When a user attempts to login to the QuickAuth authentication entity with a username, a pre-installed application that is running on both the user's mobile device and computer are automatically triggered with a remote request. The applications open each device's built-in microphone and synchronously records the room's ambient sound. If the recorded ambient sounds collected by the mobile device and the computer match, the login succeeds; otherwise,

the login fails. This system may not fit all environments because it depends on an ability to listen to ambient sound. Nearby audio may be recorded, including any personal conversations occurring in the background, which could lead to security vulnerabilities and infringe on the privacy of individuals. The user may also be unable to find an environment with noise levels appropriate for the authentication system to reliably work [44]. If there is a single loud source of noise (e.g., a television, nearby construction, music, etc.), it may also be possible to trick the system into thinking that two mobile devices are in the same location [57].

Similar to the previous example in which ambient audio captured simultaneously by two devices was used, Wang, Zhu, Yan, and Wang published a system called Sound Auth that generates random, near-ultrasound noise for more accurate comparison [58]. In this system, a user initiates a login request on a remote authentication entity, and both devices begin recording. Both the background noise and the near-ultrasound noise that is generated by the user's web browser are recorded for the duration of the login procedure. When the recording is completed, each device sends the recorded audio to the authentication entity for comparison to determine whether to accept or deny access. As with the previous example, gathering ambient sound information can be inconvenient. The possibility of overhearing important, private information could lead to significant security vulnerabilities or infringe on the privacy of users [59].

2.2 Conclusion

In the related works discussed, each introduced a method for implementing 2FA/DFA. Weaknesses were discussed that will be resolved in our proposed system. There are even a few methods that require limited user interaction, a requirement of our

proposed system. Taking account of the weaknesses of existing research, in the next chapter we propose a direction that has the potential to address these shortcomings and produce a better 2FA/DFA system that operates under our specifications.

CHAPTER 3

THE PROPOSED SYSTEM

In this chapter, we introduce our proposed system, a zero-effort 2FA system (0E2FA). To be clear, this chapter outlines the proposed system at a high level, primarily defining goals, constraints, components, and so on. We discuss the portions of the proposed system in much more detail in subsequent chapters. We also discuss some of the challenges encountered during its development and address the shortcomings that have been mentioned in the various related works in the previous chapter.

3.1 The Proposed System's Architecture

Conventional access control systems rely on users' identities to determine whether to grant access to requested resources – or not. In the proposed system, the granting of access is determined by the physical location of a user and his/her two devices (a login machine – such as a desktop computer – and a mobile device – such as a mobile phone). In fact, access is contingent upon characteristics in the user's immediate environment; specifically, characteristics of nearby IEEE 802.11 Wi-Fi devices. The second layer of security in the proposed system requires the two devices that belong to a user to be in the same physical location (more details are provided in *Section 3.3.5*). For example, if a user wishes to access an information system using the proposed system, not only is user identity required, but the user's two devices must also be in the same location. Once a user is verified, access is granted to a requested resource.

The proposed system requires a combination of the users' identity and characteristics in the users' immediate environment (specifically, characteristics that are captured by the user's two devices) to decide whether to grant or revoke access to the system.

The proposed system is also responsible for registering and validating the user's credentials before authorizing access to requested resources. The registration and authentication process is shown as:

1. A user submits his/her identification details (e.g., a username) to the authentication entity through a login machine (e.g., a desktop, a laptop, etc.).
2. The proposed system then responds with a request for additional credentials from the user to prove their identity. These credentials rely on a number of factors only known to the user and are used by the system to prove the user's identity. In this study, the proposed system depends on two user-defined factors to authenticate the user. The first factor is a description of an item only known to the user (i.e., what the user *knows*), while the second factor is a description of the user's environment – specifically, a Wi-Fi *footprint*.
3. The user sends the credential information, which includes data about the user's environment.
4. The proposed system verifies the received credential information and determines if the two devices are in the same physical location. Based on this information, a decision of whether to grant or deny the user access to requested resources is made. Access to a resource can also be granted according to a predefined set of policies.

The registration and authentication process of the proposed system at a high level is illustrated in Figure 3-1.

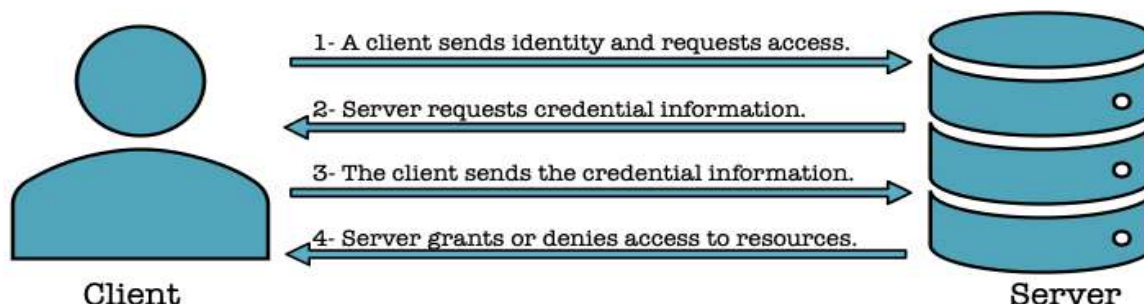


Figure 3-1: The proposed system registration and authentication processes

3.2 The Proposed System's Design Specifics

In this section, the proposed system's design is presented. Moreover, the system was designed to overcome the weaknesses of other systems noted in the previous chapter. These will also be discussed. The proposed system utilizes IEEE 802.11 (Wi-Fi), which meets various design requirements; specifically, it is inexpensive and broadly available [60]. This results in a 2FA system that can be deployed immediately in virtually any existing setting. It is also intrinsically scalable, which makes it usable in today's environment.

3.2.1 Requiring Zero-Effort to Implement Second Layer of Authentication

The objective of this research is to design a zero-effort 2FA system that uses the signals broadcast by Wi-Fi access points as the second authentication factor – without requiring user intervention (which directly addresses user convenience). As mentioned in the technical background section, each access point periodically broadcasts a unique beacon frame to all network interface devices within range. In the proposed system, both the login device and the user's mobile device collect the broadcast beacon frames from

nearby access points and send this information to the authentication entity which uses it to authenticate the user. The principal benefit of collecting embedded information from the beacon frames is that a wireless client can read and process the embedded information without necessarily being connected to the corresponding access point (e.g., for network access).

In the proposed system, IEEE 802.11 (Wi-Fi) is used as the medium because of its ubiquity. After all, wireless networks have been continually growing in popularity and can be found nearly everywhere [60]. In addition, similar to several of the systems mentioned in the previous chapter, devices that have Wi-Fi capabilities (such as mobile devices) are readily available for use as challenge tools in authentication systems because almost every internet user in the world owns one. As of 2018, according to the Pew Research Center, 96% of Americans own a smartphone, and 81% of them use their smartphones as the primary method for online access. This is especially common among younger adults [1]. In addition, nearly three quarters of American adults own desktops or laptops, while approximately half own tablets [1].

3.2.2 Verification Processes

The proposed system is a type of access control system. Access control is a security technique that provides access to a place or a resource based on selective restrictions that have been described by some administering entity. As an access control system, the proposed system manages identification, authentication, and authorization (see Figure 3-2).

Identification is a logical entity that is used to prove a user's identity on a system. In the proposed system, the identification phase can be carried out by requiring a user to

type in a username using relevant applications (e.g., the login machine and mobile device applications) that are used as the means by which a user is identified.

Authentication verifies a user in relation to a provided identity. This stage can be executed when a user further provides a correct password, for example (i.e., one that is associated with the provided username).

Authorization is an act that validates data according to a predefined mechanism and, based on the result, either gives permission or not. Here, it is performed by the proposed system's authentication entity, which compares the information received from two devices (something that is *in the user's environment*): the one used to attempt the login with the remote service and the user's pre-registered mobile device.

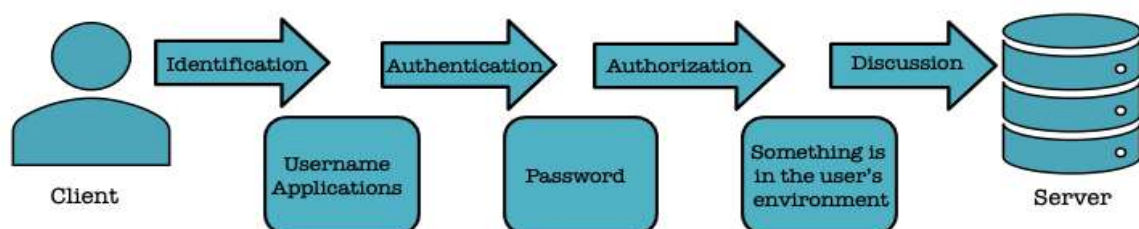


Figure 3-2: Verification processes

3.2.3 The Proposed System's Components

The proposed system requires three components: (1) a remote authentication entity (which could include the users' requested resources); (2) a login machine (e.g., a laptop, desktop, etc.); (3) a mobile device (e.g., a cellphone).

The authentication entity can be local or remote with respect to the network. The local authentication entity can be in a user's local area network (LAN); the remote authentication entity can be on the internet or some other wide area network (WAN), for example. However, the user has the same degree of control whether it is in the same

location as the authentication entity or not (i.e., this characteristic does not matter). The proposed system's authentication entity is effectively the "brain" that implements the proposed system and controls access to requested resources.

In the proposed system a user logs in, and the system automatically collects what is in the user's environment (i.e., the Wi-Fi footprint that this device "sees"). A mobile device therefore becomes an authentication device. Using the mobile device, the system authenticates the user's identity and also collects what is in the user's environment (again, the Wi-Fi footprint that this device "sees").

3.2.4 Database Design

In general, a database is a systematic collection of data that is stored and organized electronically on a computer system. The proposed system requires a database to store and retrieve information when needed. The proposed system receives information in a bag form (more details are provided in Section 3.4.1) then stores the information in a relational database. This makes it easy to find specific information and also to sort information based on any field and table, as shown in Figure 3-3.



Figure 3-3: A simplified Entity Relationship (ER) diagram of the relational database used for the proposed system.

In the proposed system's database there are five tables, and each table has a set of fields. The User table contains the information of users (e.g., id, name, username, password, email, registration date, and last visit date), as shown in Figure 3-4.

Name	Type	Collation	Attributes	Null	Default	Extra
id 🔑	int(11)			No	None	AUTO_INCREMENT
name 🔑	varchar(400)	utf8mb4_unicode_ci		No		
username 🔑	varchar(150)	utf8mb4_unicode_ci		No		
password	varchar(100)	utf8mb4_unicode_ci		No		
email 🔑	varchar(100)	utf8mb4_unicode_ci		No		
registerDate	datetime			No	CURRENT_TIMESTAMP	
lastvisitDate	datetime			No	CURRENT_TIMESTAMP	

Figure 3-4: Structure of the user's table in the proposed system's database

The mobile device information table contains the mobile device information of registered users (e.g., user id, IMEI, device id, and session status (more details are provided in Section 3.4.1)) is shown in Figure 3-5.

Name	Type	Collation	Attributes	Null	Default
user_id  	int(11)			No	None
imei	varchar(100)	latin1_swedish_ci		Yes	NULL
androidID	varchar(50)	latin1_swedish_ci		Yes	NULL
session status	int(11)			No	0

Figure 3-5: Structure of the mobile device information table in the proposed system's database

The Mobile Data and Login Machine tables contain data that is collected by both of the user's devices (e.g., user id, scan time, SSID and BSSID of every access point in the area – the Wi-Fi footprint, and RSSI value (i.e., that is in the table called level)) as shown in Figure 3-6.


Name	Type	Collation	Attributes	Null	Default
user_id 	int(11)			No	None
scan_time	timestamp			No	CURRENT_TIMESTAMP
ssid	varchar(100)	latin1_swedish_ci		Yes	NULL
bssid	varchar(100)	latin1_swedish_ci		No	None
level	varchar(100)	latin1_swedish_ci		Yes	NULL

Figure 3-6: Structure of the collected data by mobile device table in the proposed system's database

The Overlapping table contains the overlapping access points between a user's login machine and mobile device (e.g., user id, SSID and BSSID of the overlapping access points in the area, and RSSI value of the mobile device, and the RSSI value of the login machine) as shown in Figure 3-7.

Name	Type	Collation	Attributes	Null	Default
user_id 	int(11)			No	None
ssid	varchar(100)	latin1_swedish_ci		Yes	NULL
bssid	varchar(100)	latin1_swedish_ci		Yes	No
m-level	varchar(100)	latin1_swedish_ci		Yes	NULL
d-level	varchar(100)	latin1_swedish_ci		Yes	NULL

Figure 3-7: Structure of the overlapping access points table in the proposed system's database

Together, the tables in the relational database allow the authentication entity to manage users and their access to requested resources.

3.2.5 Determining the Location of a User's Devices

The proposed system utilized only SSID, BSSID, and RSSI to determine a user's location because (1) Using less will decrease the proposed system's accuracy. In Chapter 5 Section 5.1, it was determined that the RSSI readings were not consistent and also using RSSI values alone does not provide enough information when trying to determine a user's location. Regarding the SSID, there is a possibility that two access points contain the same SSID. Therefore, the BSSID is included because it is a unique identifier of an access point. Although it is possible that two access points have the same BSSID (e.g., through spoofing of the MAC address), this is very rare.; (2) Utilizing more will not increase the proposed system's accuracy as it was determined in Chapter 5 Section 5.2.

The proposed system is intended to be a zero-interaction system such that a user does not need to intervene during the authentication processes (other than providing an identity and the first authentication factor – such as a password). This model is unique in that it grants access only when the two devices are determined to be in the same vicinity. Vicinity is determined by the Wi-Fi footprints of the user's devices. If the two devices

obtain the same Wi-Fi characteristics in the environment (e.g., SSID, BSSID, RSSI), an assertion of co-location can be made. This allows the system to remove what we believe to be a significant “weak link” from other 2FA systems: user inconvenience. Specifically, this allows a user to log in with the benefits of 2FA, but without the usual hassle involved with it (e.g., manually typing in a PIN, phrase, or physical key-based second factor). Most users are familiar with and unencumbered by SFA, making the initial typing of the username and password the only action that they are required to perform. We believe that users will find this method as convenient as SFA, which would encourage its use and enable deployment.

Verifying that both devices are in the same physical location can be done using information extracted from the Wi-Fi footprints provided by the user's devices. The presence of similar information from all nearby Wi-Fi access points acts as the first layer of location verification. From there, calculations using RSSI values revise the distance between each device and the access point in the area. Finally, these measurements are compared to determine if both devices are in the same physical location. Based on this outcome, the authentication entity either allows or denies access to requested resources.

3.3 Operation

In this section, we describe the operation of the proposed system that takes place in two phases: (1) the registration phase; and (2) the login and authentication phase. Because the proposed method utilizes ubiquitous IEEE 802.11 Wi-Fi access points, it can be easily and inexpensively implemented [60]. Without requiring any interaction from the user, the method can authenticate and authorize access. The proposed system contains three elements:

1. The login machine application: When a user is finished with the registration phase, the login machine must have a custom application installed. The login machine application is used to attempt a login to the remote authentication service as shown in Figure 3-8. Through it, the user's computer (or similar device) collects received signals from different access points in the area and sends them to the remote authentication entity automatically.

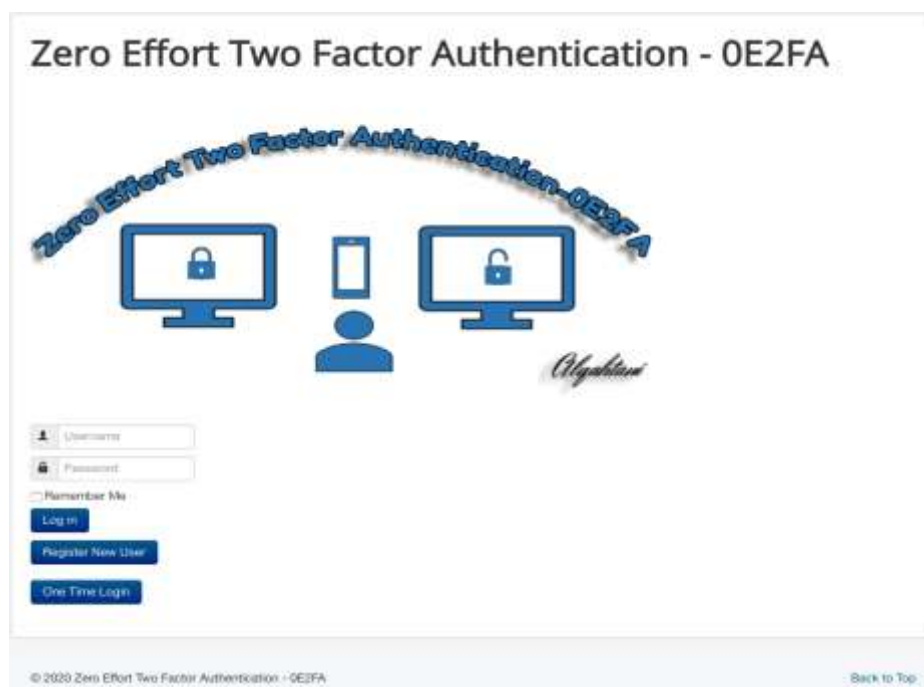


Figure 3-8: Login machine application

2. Mobile device application: A user has to download the proposed system's mobile device application during the registration phase (see Figure 3-9). Through the application, the user is able to activate their account (see Figure 3-10). Because the proposed system is intended to require zero interaction on the part of the users, the mobile device's application that automatically launches and manages the login process on the mobile device must be installed during the registration

phase. This application will automatically scan and collect data from surrounding Wi-Fi access points for authentication requests. The application ultimately sends the collected data to the authentication entity.



Figure 3-9: Mobile device application



Figure 3-10: Activation of an account

3. The authorization entity: The authorization entity receives data in a Bag form (details of this are provided in the following section) from users and stores this in a relational database. Data received from users during the authentication phase is utilized in order to make the decision to either grant or deny access to the requested resources.

3.3.1 Communication Overview

As shown in Figure 3-11, the proposed system is set up in a client-server architecture, with arrows indicating the direction of communication. The user's devices (mobile device and personal computer) require network access to communicate with the authentication entity.

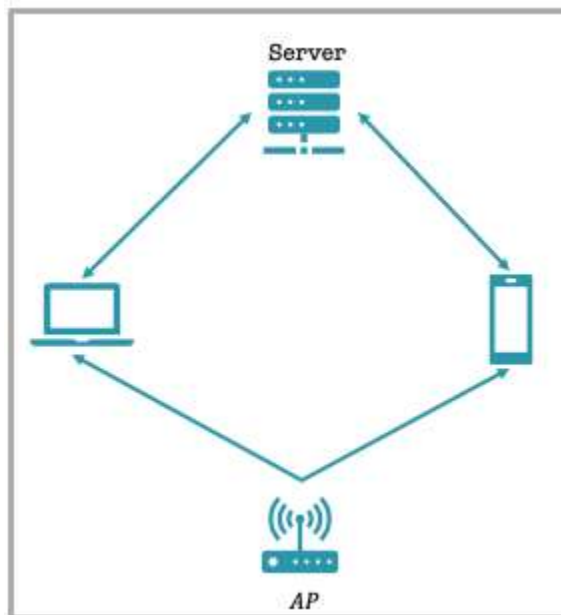


Figure 3-11: Communication system overview

The mobile device can be connected to the authentication entity either through a cellular data network or a Wi-Fi connection, while the personal computer (or similar

device) can be connected to the authentication entity via Wi-Fi, Ethernet, or similar. The broadcast messages of access points in the area can be detected by network interface devices without necessarily establishing communication with the access points (i.e., the network devices simply need to monitor wireless broadcast messages). The communication between the authentication entity and the user's computer is a two-way communication (i.e., from the authentication entity to the computer—and vice versa). This communication occurs over the Hypertext Transfer Protocol (HTTP), which is an application layer protocol that is used as a request-response protocol between a client and the authentication entity.

The communication between the authentication entity and the mobile device is also a two-way communication (i.e., from the mobile device to the authentication entity—and vice versa). This communication also occurs over HTTP by utilizing the Apache HttpClient library, which is a transfer library that resides in the mobile device application and handles sending and receiving HTTP messages between a client and authentication entity.

When a user submits credential information to the authentication entity through the login machine, the credential information travels across the internet in a form of a network packet (i.e., a unit of data that flows from a source to a destination). After verifying the user's credential information, the authentication entity will then proceed to change the user's status to an active state within the database. The devices will then be instructed to collect the beacon frames within an area of each device as soon as this user's status flag is set to active. The mobile device application will continuously check the status of a user to ensure that the user is in an active state, these checks are happening as

a background process that is initialized once the user has been authenticated. This background process is in constant contact with the authentication entity to check that the user is still authenticated and active while also not requiring the application to be visibly running. Then both devices send the requested information (i.e., Wi-Fi footprint, RSSI values, etc.) back to the authentication entity after packing them for transmission. The requested information will be grouped in a *Bag*, which is a set of n tuples (i.e., an ordered list of n fields) where n is the number of access points in an area.

$$Bag = \{AP_1, AP_2, \dots, AP_n\} \quad n \geq 1 \quad (\text{Eq. 3-1})$$

In the proposed system, each tuple is made of three fields that contain the data received (i.e., SSID, BSSID, RSSI) from an access point.

$$Bag \left\{ \begin{array}{l} Tuple_1 = \{SSID_1, BSSID_1, RSSI_1\} \\ Tuple_2 = \{SSID_2, BSSID_2, RSSI_2\} \\ \vdots \\ Tuple_n = \{SSID_n, BSSID_n, RSSI_n\} \end{array} \right. \quad (\text{Eq. 3-2})$$

The login machine's application receives a decision from the authentication entity to authenticate and authorize the user as discussed earlier. Note that a discussion of the communication security and how the beacon information is structured appears later in this text.

In the proposed system, the authentication entity is effectively the controller, in that after it receives the data from the client side, all operations are controlled and implemented by the authentication entity. It is in charge of verifying the credential information of the user. Furthermore, it hosts the various user profiles – which, if desired, could be located on an entirely separate database server behind a firewall (i.e., a network system that controls the network traffic based on predefined rules including inbound,

outbound traffic, port information, and type of traffic, for the purposes of additional security. Lastly, the authentication entity is configured with the required access policies, and based on them, a decision to authenticate and grant permission to access requested resources is made.

3.3.2 Implementation

The proposed system consists of an authentication entity that is configured with an access policy. At the other end are the client-side applications, which are designed to collect the beacon frames broadcast by Wi-Fi access points in the area, and subsequently submit them to the authentication entity. Only some of the information within the beacon frame is utilized to authenticate and authorize the user at the authentication entity. Figure 3-12 presents an overview of the proposed system's operation, which is followed by a step-by-step explanation.

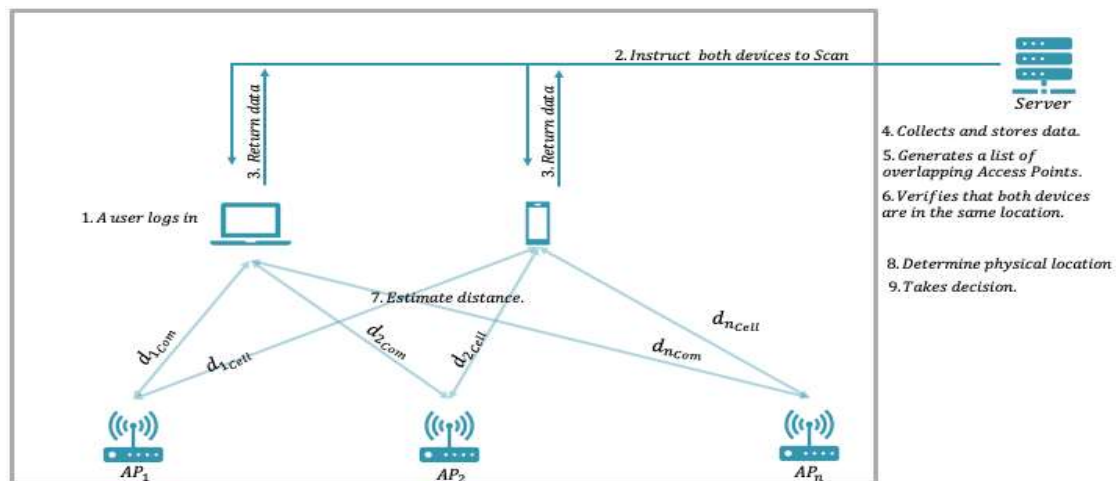


Figure 3-12: Overview of proposed system

1. A user requests access to resources at the authentication entity and submits login credentials to the authentication entity through the login machine's application

that is installed during the registration phase. The authentication entity immediately verifies the received credential information (i.e., by running a hash function on the backend for a user's password then comparing the username and hashed password with ones that were setup and stored in the database during the registration phase).

2. Every access point in the area periodically broadcasts its own beacon frame. After verifying the user's credential information, the authentication entity instructs both devices to collect their respective Wi-Fi footprint measure the RSSI value of every presented access point (more details are provided in Section 3.4.1). The authentication entity then triggers the mobile device application when the verification of the credentials is successful (more details are provided in Section 3.4.1). Both devices send the requested information (e.g., Wi-Fi footprint, RSSI values, etc.) back to the authentication entity (more details are provided in Section 3.4.1).
3. The authentication entity receives and stores the records in a relational database (more details are provided in Section 3.3.4 and 3.4.1).
4. From the data provided, the authentication entity generates a list of overlapping access points (those detectable by both devices) by checking the SSID and BSSID of each (more details are provided in Section 3.3.5). This list will be used for all authentication steps from here on. As mentioned in the previous section the data is sent to the authentication entity in a *Bag*. In this phase, the authentication entity receives two bags (one from the login machine (*Bag_{Login machine}*) and the other bag from the mobile device (*Bag_{Mobile}*)).

$$Bag_{Login\ machine} \begin{cases} Tuple_1 = \{SSID_1, BSSID_1, RSSI_1\} \\ Tuple_2 = \{SSID_2, BSSID_2, RSSI_2\} \\ \vdots \\ Tuple_n = \{SSID_n, BSSID_n, RSSI_n\} \end{cases} \quad (Eq. 3-3)$$

$$Bag_{Mobile} \begin{cases} Tuple_1 = \{SSID_1, BSSID_1, RSSI_1\} \\ Tuple_2 = \{SSID_2, BSSID_2, RSSI_2\} \\ \vdots \\ Tuple_n = \{SSID_n, BSSID_n, RSSI_n\} \end{cases} \quad (Eq. 3-4)$$

Then the authentication entity determines the overlapping access points by comparing the $SSID_i$ and $BSSID_i$ from the two bags. The authentication entity verifies that both devices are in the same location (more details are provided in Section 3.3.5), checking the list of overlapping access points. Figure 3-13 shows an example of a list of overlapping access points.

Wifi Name	BSSID	Mobile				Desktop			
		First Distance	Last Distance	Difference	Difference (%)	First Distance	Last Distance	Difference	Difference (%)
Access Point is within Range (2 m) from Mobile and Desktop									
ALJ	1c:49:7b:dd:40:75	9.5	9.5	0	0	11.19	11.19	0	0
ALJ5G	1c:49:7b:e0:61:38	11.58	11.58	0	0	11.88	11.88	0	0
AP1	10:da:43:c2:09:19	8.61	8.61	0	0	9.42	9.42	0	0
AP15G	10:da:43:c2:09:16	9.52	9.52	0	0	8.91	8.91	0	0
Mobile(Min 5%)					Desktop(Min 5%)				
015caa	bc:2e:48:6c:0f:9e	25.31	25.31	0	0	31.54	31.54	0	0
DIRECTq7C460 Series	32:cd:a7:1c:ae:1e	5.03	5.03	0	0	2.37	2.37	0	0
DroptLikeltsHotspot	8c:3b:ad:e6:94:92	29.32	29.32	0	0	44.55	44.55	0	0

Figure 3-13: An example of a list of overlapping access points.

5. The RSSI readings are then analyzed using Equation 3-1 to estimate the distance between the user's two devices and every access point "seen" in the area.

Equation 3-1 is the log-distance path loss model that predicts a path loss of a signal inside a building over a distance. It is used in the proposed system to estimate the distance between a transmitter and a receiver.

$$PL_{log} = PL_0 + 10 \gamma \log_{10} \frac{d}{d_0} \quad (\text{Eq. 3-5})$$

where

PL_{log} is the transmitted power (i.e., 20) minus the received power (RSSI value)

$$PL_{log} = P_{tx} - P_{rx} \quad (\text{Eq. 3-6})$$

PL_0 is the path loss in dBm (PL means that there is a reduction in power strength as a signal travels through space).

$$PL_0 = 20 \log_{10} \left(\frac{4\pi d_0}{\lambda} \right) \quad (\text{Eq. 3-7})$$

λ is the operating wavelength in meters (e.g., for 2.4 GHz is approximately equal 0.125 meters and for five GHz is approximately equal 0.06 meters),

γ is the path loss exponent (PLE) in dBm (PLE is a parameter in wireless communications that represents the propagation of fading channels).

d_0 is the reference distance between the transmitting and receiving antenna in meters (usually equal to one meter).

For example, if the RSSI value was -35 dBm the distance between the device and the access point is calculated as:

$$20 - (-35) = 20 \log_{10} \left(\frac{4\pi * 1}{0.125} \right) + 10 * 2.7 \log_{10} \frac{d}{1}$$

$$55 = 40.46 + 27 \log_{10} d$$

$$55 - 40.46 = 27 \log_{10} d$$

$$14.54 = 27 \log_{10} d$$

$$d = 10^{\frac{14.54}{27}}$$

$$\therefore d = 3.46 \text{ meters} \quad (\text{Eq. 3-8})$$

6. The distance between an access point and the user's devices is used to refine the determination of whether the two devices are in the same physical location or in a neighboring physical location (more details are provided in Section 3.3.5). Figure 3-14 shows an example of the distances (in meters) between a user's devices (i.e., login machine, mobile device) and four access points (noted by red boxes in the figure).

Wifi Name	BSSID	Mobile				Desktop			
		First Distance	Last Distance	Difference	Difference (%)	First Distance	Last Distance	Difference	Difference (%)
Access Point is within Range (2 m) from Mobile and Desktop									
ALJ	1c:49:7b:dd:40:75	9.5	9.5	0	0	11.19	11.19	0	0
ALJ5G	1c:49:7b:e0:61:38	11.58	11.58	0	0	11.88	11.88	0	0
AP1	10:da:43:c2:09:19	8.61	8.61	0	0	9.42	9.42	0	0
AP15G	10:da:43:c2:09:16	9.52	9.52	0	0	8.91	8.91	0	0
Mobile(Min 5%)					Desktop(Min 5%)				
015caa	bc:2e:48:6c:0f:9e	25.31	25.31	0	0	31.54	31.54	0	0
DIRECTq7C460 Series	32:cd:a7:1c:ae:1e	5.03	5.03	0	0	2.37	2.37	0	0
DroptLikeltsHotspot	8c:3b:ad:e6:94:92	29.32	29.32	0	0	44.55	44.55	0	0

Figure 3-14: An example of the distances between a user's devices and four access points which are identified by red boxes.

7. A decision is made to allow or deny access to the user based on the shared location and proximity of the two devices (more details are provided in Section 3.3.5). Figure 3-15 shows an example that the user has been successfully authenticated after all the steps have been met successfully.

Zero Effort Two Factor Authentication - 0E2FA

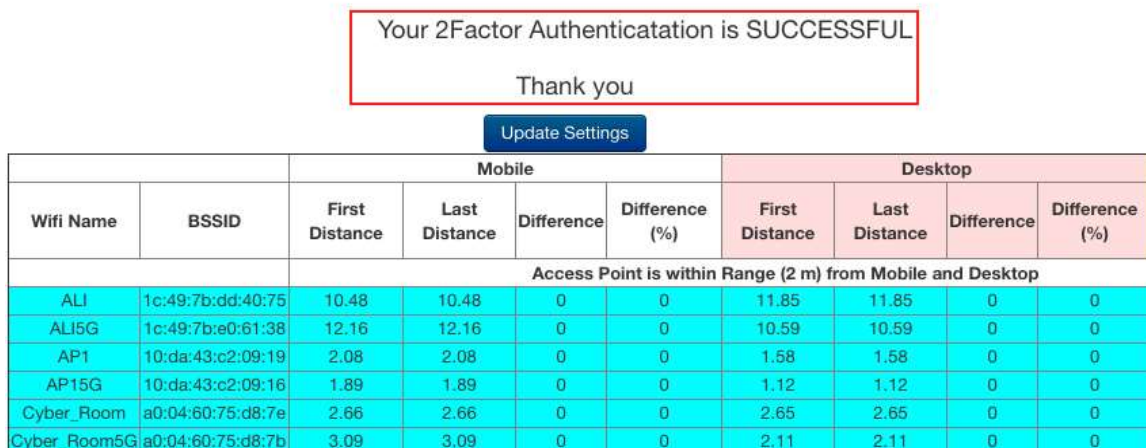


Figure 3-15: A diagnostics screen for testing purposes.

The authentication entity instructs the user's devices to collect what is in the user's environment (i.e., SSID, BSSID, and RSSI value) and uses the collected information to authenticate the user (more details are provided in Section 3.4.1). A set of rules below guides this authentication process:

- a. There exists at least a predefined number of overlapping access points detected by both devices; and
- b. The estimated distance from the devices to all access points falls within a predefined range. However, if there are no overlapping access points, the user can gain access using a one-time login feature that will be discussed in Chapter 4 Section 4.3.2.

Because the proposed method utilizes ubiquitous IEEE 802.11 Wi-Fi access points, it can be easily and inexpensively implemented in a real-world environment [60]. The proposed system allows flexibility with respect to authentication and authorization as it can implement these two phases without placing extra burden on users. The proposed

system requires no more intervention than merely entering the username and password as usual in a SFA system. The proposed system therefore has the convenience of SFA with the security of DFA, without the vulnerabilities associated with only having SFA and the inconvenience of typical DFA systems.

3.4 The Proposed System's Features

A study of the existing 2FA systems from different perspectives was performed prior to the start of the design phase of our system. Both the positive and negative aspects of each system was considered in order to design a novel 2FA system that addresses the various shortcomings mentioned in Chapter 2. As a result, the proposed system is effective as a DFA system that reduces impact on users. Moreover, it has several distinctive features: (1) security; (2) user convenience; (3) flexibility; (4) cost effectiveness.

3.4.1 Security

Security is one of the most essential features of the proposed system. In general, security is an approach to protect valuable possessions against potential threats, either visible or invisible. In information systems, security is a defense technique for digital information against either internal or external malicious activities. Security is implemented in three forms: (1) detection, (2) prevention, and (3) response to threats. In the proposed system, all three forms were considered to ensure that it is resistant to unauthorized access and against potential attacks. A discussion of this is provided in Chapter 4.

3.4.2 User Convenience

User convenience typically describes how a product or service is conducive to achieving user satisfaction. In the security area, the relationship between security and user convenience is an inverse relationship. In other words, as security increases, user convenience often decreases—and vice versa. In this work, the various aspects of the proposed system were carefully designed with the aim of balancing user convenience with system security. This is further discussed in Chapter 4.

3.4.3 Flexibility

A system is flexible if it can accept one or more modifications to achieve a desired fit or configuration. For example, the proposed system has the ability to adjust the number of access points that an administrator may desire the system to use, among other things. Using more access points may provide a more accurate determination of the co-location of a user's devices. A discussion of this, including more details, is presented in Chapter 4. In this chapter, we merely layout the intended characteristics of the proposed system.

3.4.4 Cost-Effectiveness

The term cost-effective means that a system has value when its benefits are compared with its cost. The relationship between a system's costs and benefits may be a positive relationship, where if one goes up, the second follows—or the relationship may be negative, which is the inverse. In the proposed system, what a user *possesses* (i.e., a mobile device and a larger computer such as a desktop or a laptop) and what is in the user's *environment* (i.e., the Wi-Fi footprint) are used. Since the proposed system makes use of devices that users and entities within the environment already have, there are no

additional costs incurred to implement it. In addition, no specialized hardware needs to be installed or used. Ultimately, the proposed system is relatively easy to deploy and maintain.

3.4.5 Logging in Using a Login Machine

There is a case where the user may not have the authentication device in possession (e.g., the device was forgotten, lost, or stolen) or possesses a device that does not have the ability to scan for Wi-Fi networks (i.e., does not have an internal wireless network interface). The proposed system manages this problem by allowing the user to use a one-time login feature. More details will be provided further in this text that explain this and similar challenges.

3.5 Challenges

For the proposed system to meet the design requirements of an effective and efficient 2FA system (as we have previously defined), some elements in the proposed system must be designed carefully in order to achieve the stated goals. Below are some of the challenges of achieving such a system.

3.5.1 Collecting Data from a User's Mobile Device

Collecting data from the user is an essential and significant aspect of the proposed system. In our proposed system, a mobile device is required to scan for and analyze the beacon frames of Wi-Fi access points within range; furthermore, the measurement of the RSSI values that indicate a distance to each access point in the area must be securely sent to the authentication entity. Because the system is intended to require no interaction on the part of users, an application that automatically executes when necessary and manages the login process on the user's mobile device must be implemented. The application must

automatically scan and collect data from surrounding Wi-Fi access points, along with their RSSI values, and then send the collected information to the remote authentication entity.

3.5.2 Collecting Data from a User's Login Machine (or Similar) Computer

A user's login machine (or similar computer) is designated as the terminal used to access the proposed system; specifically, it is through it that a user attempts to login to a desired resource. Similar to the mobile devices, a login machine's application is needed to scan and collect all of the visible beacon frames from Wi-Fi access points in order to measure the RSSI values of each access point in the area, and to securely send the collected information to the remote authentication entity. In the proposed system, after the user submits SFA characteristics when attempting to login, the machine's application is responsible for automatically executing the scan and aggregating the data from all of the broadcast messages (more details are provided in Section 3.4.1). Moreover, the RSSI values are measured and the data is submitted to the authentication entity (more details were presented in Section 3.4.1). This application must be developed in order to achieve a goal of the proposed system.

3.5.3 Gaining Access

In this work, gaining access refers to a process in which access to resources is granted to a user provided that the user's two devices are physically located in the same vicinity. In the proposed system, a novel method for gaining access to a system by utilizing the beacon frames and RSSI values of all access points in the area was introduced; specifically, utilizing the overlapping access points between the user's two devices. Information is collected by a user's devices and then sent to the authentication

entity for processing. This information is then subsequently used to infer a location or distance from the access points (more details are provided in *Section 3.4.2*). By utilizing the data collected from the access points, a user's position can be determined.

3.6 Conclusion

Many service providers use a 2FA system to enhance account security [61]. People use 2FA/DFA on a daily basis for different functions, such as making payments, for banking, etc. Furthermore, standard names in industry such as Apple, Google, and Microsoft, are increasingly enforcing a 2FA system for their users [62]. The proposed system controls access to resources by utilizing what is in a user's *environment*. This is a significant contribution of this work and what distinguishes it from many other existing works. We believe that strengthening the security of 2FA systems is research-worthy. Moreover, we believe that structuring the system in such a way that minimizes user interaction increases user convenience and potentially, security as well.

In this chapter, the proposed system's high-level architecture, features, design requirements, operation, and the challenges of creating an effective 2FA system were presented. Subsequently, we proposed a high-level picture of a new 2FA system design that is intended to meet the proposed design requirements. The challenges involved in achieving such a system were then discussed. Achieving the proposed system with excellent performance is significant, because it would provide a 2FA system in which the location of the user's devices is verified in a way that guarantees authorized access to resources when the user's devices are in the same physical location, all without requiring user intervention (thereby increasing user convenience by implementing a DFA with the same user requirements as a SFA). Our proposed 2FA system does this in a convenient

way by precluding effort from users and without requiring additional infrastructure. We believe that designing an unobtrusive and inexpensive 2FA system will encourage increased integration of 2FA in access control systems in the future. The next chapter will present the proposed system's security and its response to diverse types of attacks.

CHAPTER 4

DISCUSSION

In this chapter, different aspects of the proposed system are evaluated. Moreover, the different factors and parameters that play a role in the proposed system's performance and security are analyzed. Section 4.1 will present the proposed system's security and its response to diverse types of attacks, Section 4.2 will evaluate how convenient the system is for the user, Section 4.3 will discuss the privacy of the user in the context of the proposed system, Section 4.4 will present a discussion of the robustness of the system, and Section 4.5 is a review of the proposed system's cost in a real environment.

4.1 Security

To enhance the security of the system, the proposed system implements 2FA as a solution. This way, if a user's password is compromised, the security of the system still involves another level of authentication (at least two levels of authentication in total). Furthermore, this research aims to add an extra layer of authentication in a novel way: by utilizing the characteristics in a user's *environment* – specifically, by utilizing information of ambient access points visible by the user, thus establishing a Wi-Fi *footprint*. Every access point broadcasts messages that carry distinctive information that can be used to identify a Wi-Fi footprint. Subsequently, the outcome can be used to authenticate the user. In section 4.1.1, we analyze the security features of the proposed system.

In section 4.1.2, we examine several types of anticipated attacks that could occur. We subsequently present how the proposed system responds to and addresses these attacks.

4.1.1 Security Features

In this section, the proposed system's security features are presented. Section 4.1.1.1 presents the client-side application security for the mobile device and login machine. Section 4.1.1.2 shows how the system depends on broadcast messages to limit spoofing. Section 4.1.1.3 introduces communication security and more specific information regarding what method is being used to protect network communication. Section 4.1.1.4 explains how the proposed system is adjustable, by allowing different configurations for the different thresholds. Section 4.1.1.5 presents database and infrastructure security. Finally, Section 4.1.1.6 explains the strength of the proposed system's second layer of authentication.

4.1.1.1 The client-side application security

To ensure client-side security, the mobile device's application is secured by associating it with the mobile device's IMEI and another available unique identifier of the mobile device (e.g., device ID, serial number – or both if needed), as discussed in Chapter 3, Section 3.4. The login machine's application is secured by associating it with the login machine's universally unique identifier (UUID), along with another unique identifier (e.g., serial number) if needed, as discussed in Chapter 3, Section 3.4. Without either of the applications, the user cannot use the proposed system. We realize that users may find it inconvenient to download and install the applications on their devices; however, they implement the user side mechanism that ultimately limits access to the

system and increases security. The user must have both in order to utilize the proposed system. Moreover, this is typical of authentication systems in use today that utilize mobile devices.

4.1.1.2 *Limitation of spoofing*

The proposed system is designed to be a zero-effort 2FA system that uses signals broadcast by Wi-Fi access points for use as a second level of security (a second factor) without the need for user intervention. Each access point periodically broadcasts a unique beacon frame to all devices in range. The major benefit of this is that the beacon frames can be received by a device only while it is within a limited physical range of each wireless access point. This intrinsically limits the potential for spoofing or impersonation and makes second factor authentication secure. Along with features within the beacon frame, the proposed system utilizes the measured RSSI values to calculate the distance from the various Wi-Fi access points in range to each user device. Using the RSSI value as a means to measure the user's location helps to ensure that the user is within a limited physical range of each access point that is scanned and further decreases the chances of spoofing or impersonation (i.e., RSSI value can be measured within a limited physical range of each wireless access point). The case where an attacker could spoof access points and replicate the information (e.g., SSID, BSSID, and RSSI readings) in another location will be discussed below in Section 4.1.2.3.

4.1.1.3 *Communication security*

Standard communication security is utilized to prevent an attacker from intercepting the communication between the user and the authentication entity. In the proposed system, Secure Sockets Layer (SSL) and Transport Layer Security (TLS) is

used to securely transmit information between the user and the authentication entity. Through the use of SSL/TLS, the transfer of data and information between the two entities occurs in the form of an asymmetric key encryption algorithm. An asymmetric key algorithm (also called public key cryptography) uses a pair of related keys in which one is used to encrypt and another is used to decrypt in order to protect a message from unauthorized access or an unauthorized user. It is not possible to use the same key to both encrypt and decrypt the same message. The public key exists publicly on a network such as the internet and is obtained when a user wants to securely send a message to the owner of the public key (e.g., the authentication entity of our proposed system). Only the private key (which is kept with its owner) can be used to decrypt the message. Figure 4-1 illustrates the process: (1) the user obtains the authentication entity's public key; (2) the authentication entity's public key is used to encrypt information and transform it from plaintext into ciphertext and sends it to the authentication entity; (3) the authentication entity receives the encrypted data and decrypts with its private key, allowing it to read the user's confidential message.

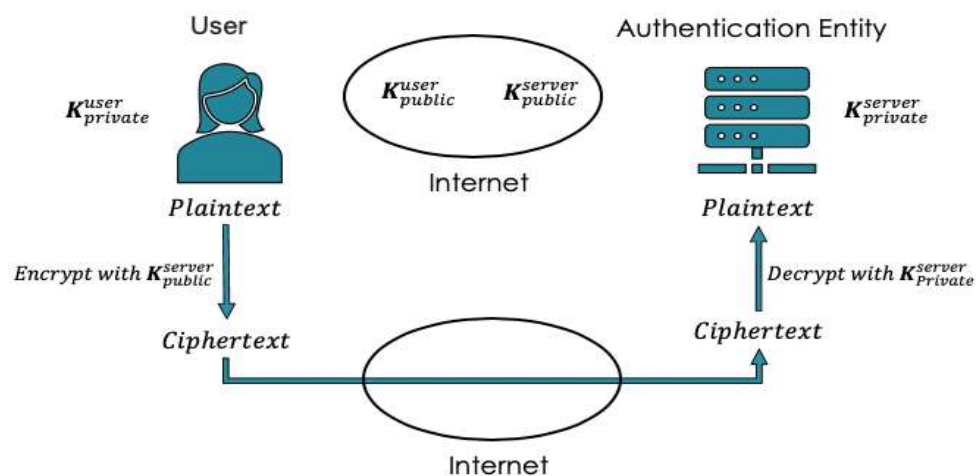


Figure 4-1: Process of an asymmetric key encryption algorithm

4.1.1.4 *Parameter adjustability*

As mentioned in Chapter 3, the proposed system is adjustable. Specifically, it has the ability to adjust the minimum number of access points to use to make an authentication decision. The scanning range of the access points can also be adjusted. If the range of the system is set to five meters, for example, the system will only use the access points that fall within that range of the user's devices. All of these features are beneficial for fine-tuning the system and increasing the security; specifically, by determining and limiting the range of the system.

4.1.1.5 *Database and infrastructure security*

In the proposed system, the database, web, and application servers are deployed in a distributed network in order to avoid a single point of failure. The storage of records in plaintext on the database server is limited by applying standard server security: (1) user passwords are combined with salt (a random string of data that is added to a user's password to minimize the chances that multiple users with the same passwords have the same data stored in the database); and (2) the user password/salt combination is hashed using a secure hashing algorithm (e.g., SHA-256 which takes variable-sized input and produces a fixed-size output). For example, the word “original” is hashed through SHA-256 hash function and becomes “25718360e05d3c2d0963d1381e9dd4dae5fca789244ee4b9f861adcc0cc96218”.

4.1.1.6 *Strong second layer of authentication*

The proposed system is designed to be secure and different from existing 2FA approaches that typically require user interaction to authenticate a user. The first authentication factor is something that a user *knows* (username for identification and

password for authentication). The second authentication factor is something unique that is in the client's *environment* (the SSID, BSSID, and RSSI features of the beacon frame).

Without requiring any interaction from the user, the proposed system authenticates and authorizes access based upon these identification and authentication factors. As explained in Chapter 5, the user gains access if the two devices meet the following criteria:

- a. The device is positioned such that a predefined number of overlapping access points is detected by both of the user's devices; and
- b. The estimated distance from the user's devices to all visible access points falls within a predefined range.

The proposed system depends on two factors to authenticate a user which increases the strength of the second layer authentication.

4.1.2 Cyberattack Vulnerability Assessment

In this section, the proposed system's vulnerability to several identified cyberattacks are assessed and discussed. Furthermore, the steps taken to address these types of attacks are also discussed. Section 4.1.2.1 focuses on identified man-in-the-middle attacks and how the proposed system is designed to preemptively avoid them. Section 4.1.2.2 focuses on eavesdropping attacks and ways to stop an attacker from gaining access to data sent over the network. Section 4.1.2.3 looks at the possible simulation of the user's environment for obtaining unauthorized access from a location outside of a legitimate user's environment and how the use of the custom application on the user's login and mobile devices can mitigate these attacks. Section 4.1.2.4 focuses on the potential for an attacker to be inside of a user's environment (i.e., in the same room or

office), and how through the use of special identifiers within the application that are installed on a device, this kind of attack is mitigated.

4.1.2.1 *Man-in-the-middle attack*

A man-in-the-middle attack occurs when an attacker is inserted in the system and secretly intercepts communication occurring between two entities within the system. In the proposed system, this would likely occur in between the user and the authentication entity. A man-in-the-middle could transmit (and possibly even alter) communication between two parties who believe that they are directly communicating with each other. This results in the appearance of normal communication between the two parties. In our proposed system, SSL/TLS is utilized for communicating between a user's devices and the authentication entity. This protocol is secure and effectively thwarts man-in-the-middle attacks because the communication between the two parties is encrypted within the channel of communication. All communication between the user and the authentication entity is encrypted with a secret key that is only known to the user and the authentication entity. In the general case, the proposed system utilizes standard asymmetric cryptography with asymmetric encryption key exchange protocol (more details were provided in *Section 4.1.1.3*) in order to mitigate man-in-the-middle attacks.

4.1.2.2 *Eavesdropping*

In this type of attack, an attacker eavesdrops on the communication between the user's device and the authentication entity in order to obtain credential information and use it to attempt to authenticate and gain access to a resource. In the proposed system, the SSL/TLS protocol is utilized to encrypt communication between a user's device and the authentication entity. A user's credential information (e.g., username and password) can

still be sniffed from network communication; however, an attacker cannot make sense of this because it is strongly encrypted.

4.1.2.3 *Simulation of the user's environment*

The proposed system utilizes what is in a user's *environment* (specifically, a Wi-Fi *footprint*) as an unobtrusive second authentication factor. In this attack, an attacker scans the access point information around the user's environment (e.g., an office) – specifically, the attacker obtains the SSID, BSSID, and RSSI of the surrounding/visible access points. The attacker then proceeds to replicate the user's environment elsewhere, where the attacker may have full control of the environment – specifically, the footprint of Wi-Fi devices in the environment. In general, an attacker has a relatively low probability of obtaining a user's credentials (e.g., username and password) [63].

Moreover, a user's environment can still be obtained; however, the attacker cannot make sense of it without the installed client applications because the proposed system is designed to establish communication only through the relevant applications. The applications also obfuscate the features required within the beacon frame that the authentication entity utilizes to identify users and make authentication decisions. However, if an attacker was able to obtain the device applications and reverse engineer them to work with his own devices, the attacker's device specifics (e.g., IMEI) will not match the user's information in the authentication entity's database. Moreover, the legitimate user will be informed of this because the authentication entity will reject the request.

4.1.2.4 *An attacker inside a user's environment*

By using a device that is Wi-Fi capable, the surrounding user's environment (specifically, the Wi-Fi footprint) can be captured. Furthermore, the RSSI values of the access points near the user can be measured. In this attack, it is assumed that an attacker has the username and password of a user (which is unlikely), while also being nearly in the same place as the user (e.g., immediately outside of the user's office). In the registration phase of the proposed system, each of the applications that are installed on the user's login machine has metadata associated with the registered user (e.g., the applications run on the user's devices which have specific values for IMEI, UUID, and so on). Once the user has registered, the IMEI of the mobile device and the UUID of the login machine are not allowed to change. Thus, an attacker would not be able to login without physically possessing the user's devices; in fact, *both* the login device and the mobile device (since they are both associated with the users). Ultimately, an attacker may be able to collect the data in a user's environment; however, they will not be able to establish communication to the authentication entity without the required applications and appropriate user devices.

4.2 User Convenience

In general, user convenience is a critical factor in determining how users make decisions about what to use [64]. The proposed system is a zero-effort system, which means that a user accesses an information system using two factors of authentication, but without the typical hassle involved or having extra steps required for the user (e.g., providing a PIN, a phrase, a randomly generated code, or a physical key-based second factor). Because SFA methods are used as the standard for authentication (i.e., users

almost always expect to type in their username and password), our proposed method is as convenient as SFA, thus making its application easy to use and readily scalable.

The proposed system is considered to be readily applicable in a scalable manner because it relies on ubiquitous Wi-Fi access points. The internet is found almost everywhere people live [65]. Due to its ubiquitous nature, the internet is an essential and robust platform for education, business, and entertainment. It has been noted that locations with reliable internet connectivity are also where access points are commonly established [60].

Accessing resources requires a user to only submit a username and a password. Subsequently, the second layer of authentication is automatically obtained and provided to the authentication entity. Moreover, the result of either granting or denying access to the user is done within a short period of time. In *Chapter 5*, an experiment was presented to show that the proposed system is capable of a seamless and quick authentication procedure.

Security and convenience are a sliding scale that requires finding the right balance between them. Security systems that are more convenient tend to be less secure, while security systems that are more secure are generally inconvenient [66]. A convenient system is a system that has an acceptable level of inconvenience for the amount of security that the system provides [66].

4.3 User Privacy

Most users login to authentication systems from various places (e.g., home, work, etc.). The proposed system scans and collects information from the user's environment (specifically, access point information such as the SSID and BSSID) – wherever that may

be – and stores this unique information in the authentication entity's database. When the user's current session is complete/terminated, all data related to the current session is purged from the database to ensure privacy by practicing minimal data collection (i.e., by limiting the storage of a user's sensitive information in the database).

4.4 Robustness

A 2FA system must be robust because failures may hinder a user's ability to utilize the system. In this section, the steps taken to enhance the operational robustness of the proposed system are discussed. Section 5.4.1 discusses the potential for the user's environment to lack access points, and Section 5.4.2 discusses the potential absence of the user's mobile device.

4.4.1 Lack of Access Points in the User's Environment

The proposed system has the ability to function with at least one access point. That is, it can operate when there is only one access point in the surrounding environment of a user – at minimum (obviously, at least one is needed). This case was tested in *Experiment 6* discussed in *Chapter 5*, and the proposed system's access policies were initially satisfied with the presence of only one access point. With this in mind, the proposed system was able to achieve a favorable outcome, meaning that a user in a remote location with possibly only one available access point still has the ability to use the proposed system and gain access to authorized resources.

4.4.2 Absence of the User's Mobile Device

As defined previously, the user's authentication device is typically a mobile device with the mobile application installed. In this case, the user may not have the authentication device (e.g., the device was forgotten, lost, or stolen) or there are no

overlapping access points. The proposed system manages this problem by allowing the user to use a one-time login solution. For this feature, the user types in a username and answers the security question shown Figure 4-2. A OTP is then sent to the registered email (see Figure 4-3), and it can be used by the user to access resources granted through the authentication entity. Admittedly, this has the side effect of requiring user interaction, which breaks a fundamental requirement of the proposed system. However, this is only used in the case that a user does not have access to the mobile device associated with the user's account (which, arguably, is the user's fault). In a typical setting, this will be quite rare.



Figure 4-2: One-time login page

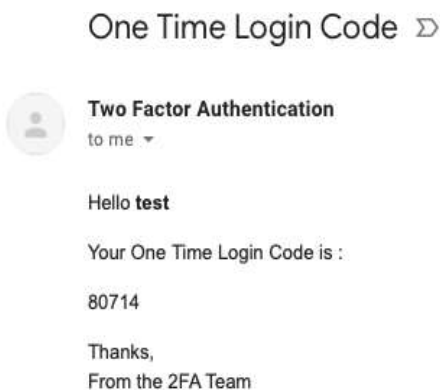


Figure 4-3: OTP

4.5 Cost

Because the proposed system utilizes existing IEEE 802.11 infrastructure (i.e., Wi-Fi access points already in the environment, network interface cards already in mobile devices and login machines, etc.), no new hardware is required. The authentication system also utilizes the personal devices of users, so no further cost is incurred either by users or organizations who wish to implement the proposed system for authentication.

4.6 Example Scenario

In this section, we discuss a potential feature of the proposed system that can extend its applicability to continuous authentication. This scenario could address a number of issues that a user may face; for example, the user forgets or leaves the login machine without logging out, potentially leaving the machine accessible to unauthorized users. Moreover, it could provide an extra level of security that an organization may wish to implement; for example, maintaining access to resources continuously (i.e., periodically checking if the user's devices are co-located instead of performing this action only once at login), or maintaining access to resources within specific environments (i.e., the Wi-Fi footprint must not only be similar for both of the user's devices, but it must also be specific in terms of SSIDs and BSSIDs – manually selecting specific Wi-Fi access points). We believe that these potential features, although not in the original scope of the proposed system, can add strength and increase relevance. A few of these features are discussed below.

4.6.1 Session Termination

The proposed system has the ability to provide continuous authentication with zero effort from the user. After access is granted, the user's devices can continuously collect and send what is in the user's *environment* (the Wi-Fi footprint) to the authentication entity for verification. The authentication entity can continuously check if the two devices are in the same location in order to maintain access to requested resources. The frequency with which the user's devices collect and send the new beacon frames can be varied based on requirements. This feature could solve the potential issue where a user has logged in to an information system and forgets to log out – or is called away. In this case, the proposed system will check if both devices are still co-located in order to keep the user's session alive. If not, the session can be automatically terminated.

An additional experiment (not mentioned in the *Chapter 5*) was performed to test how the proposed system responds when the user's devices are no longer co-located. In this experiment, the range was set to three meters (i.e., three meters was selected because the experiment's goal is to test the proposed system's response within a small range) in order to test the proposed system's response to terminating a session in such a small environment. The maximum change in distance between the user's devices across continuous readings was set to less than or equal to 5% of the previous distance, and the proposed system was set to check the difference in the distance at one minute intervals in order to avoid scanning the same RSSI value repeatedly. In the experiment, using the access points that fell within the range of three meters (AP1 and AP15G in Figure 5-4 below), the system was able to grant the user access based on operation of the system described earlier. After access was granted, the user moved to another room with his

authentication device. Once the proposed system detected that the user's devices were no longer in range (after the predetermined interval of one minute), the session was immediately terminated. This is shown in Figure 4-4. Twenty trials were implemented, and the system achieved a 100% success rate in terminating the user's session once the user's devices were not co-located.

In Figure 4-4, the information that is beneath the words *Wifi Name* represents the SSID of all the access points in the area. The *BSSID* column represents BSSID of all the access points in the area.



Figure 4-4: Session termination

Also, in the mentioned figure (Figure 4-4) the data that is beneath the column category labeled as *Mobile* is the data that is obtained from the mobile device. The *First Distance* column represents the first calculated distance between the mobile device and an access point in meters when the user first logs in. The *Last Distance* column represents the last calculated distance between the mobile device and an access point in meters of the last scan. The *Difference* column represents the difference between the *First Distance*

and *Last Distance* in meters. The *Difference (%)* column represents the difference between the *First Distance* and *Last Distance* as a percentage.

Also, in Figure 4-4 the data that is underneath the category Desktop, is the data gained from the login machine. The *First Distance* column represents the first calculated distance between the login machine and an access point in meters when the user first logs in. The *Last Distance* column represents the last calculated distance between the login machine and an access point in meters. The *Difference* column represents the difference between the *First Distance* and *Last Distance* in meters. The *Difference (%)* column represents the difference between the *First Distance* and *Last Distance* as a percentage.

4.7 Conclusion

In this chapter, different aspects of the proposed system were evaluated in the context of security, user convenience, user privacy, system robustness, and cost. Moreover, the proposed system's vulnerability to several identified cyberattacks were assessed and discussed. Based on results and discussions in this chapter, the proposed 2FA system can be considered secure, convenient, robust, and inexpensive. It utilizes existing IEEE 802.11 infrastructure, the personal devices of the users, and no new hardware is required; therefore, no further costs are incurred. The next chapter presents numerous experiments that were utilized to test the proposed system's design in various aspects.

CHAPTER 5

EXPERIMENTS

In the previous chapter, we discussed the design specifics of the proposed system and the challenges faced during the design process. This section presents numerous experiments that were utilized to test the proposed system's design in various aspects. Experiment 1 will test the location accuracy achievable from using the RSSI in the beacon frame of nearby access points obtained by network devices. Experiment 2 will examine the proposed system's authentication accuracy in terms of overlapping access points. Experiment 3 will test the RSSI behavior when using different cellphone models. Experiment 4 will show the computing and the communication cost of the proposed system (i.e., computing the time interval between the user completing a login attempt, and the authentication entity delivering an authentication decision). Experiment 5 will present the performance of the proposed system when using only one access point to perform authentication. Finally, Experiment 6 will examine the security of the mobile device application. For the experiments, the following hardware was utilized:

1. Access points: In Experiment 1, we used three Netgear R6400 Wi-Fi access points to test the location accuracy within a predefined area (see Figure 5-1). However, for the other experiments, all of the access points utilized were publicly findable in the area (i.e., they were a part of the existing production infrastructure within which the testing environment was located).

2. Authentication entity/resource server: For the authentication entity (which also included the user's requested resources), a desktop with an Intel® Xeon® processor, 16 GB of RAM, and Ubuntu 16.04 LTS (64-bit) as the operating system was utilized.
3. User devices: For the user devices, a MacBook Pro (13-inch, mid-2012 model) with a 2.5 GHz dual-core Intel Core™ i5 CPU and eight GB 1600 MHz DDR3 was used as the mobile device in the first experiment. For the remaining experiments, the following mobile devices were used: (1) Samsung Google Nexus S; (2) Samsung Galaxy S9; and (3) Samsung Galaxy S6. These were used to test the proposed system using different models of mobile devices. In addition to using a MacBook Pro, a Precision 7820 Dell desktop equipped with an Intel Xeon processor, 15 GB of RAM, Ubuntu 16.04 LTS (64-bit) as the operating system, and a Wi-Fi adapter (Alfa AWUS036NH) connected via USB was utilized as the user desktop login machine.



Figure 5-1: Netgear R6400 Wi-Fi router [67]

5.1 Experiment 1: Testing Location Accuracy in a Predefined Area

This experiment was conducted to determine the degree to which RSSI values obtained from network devices can be used to calculate a device's location. In this experiment, we utilized three access points at different locations in a room (see Figure 4-2). The first access point was set up at (0,0), the at (9, 9), and the third at (0, 18)). The room is a typical office space that is approximately 18-ft. by 9-ft.

Verifying that both devices are in the same physical location can be done using information extracted from the Wi-Fi access point broadcast messages. The presence of similar information from all nearby Wi-Fi access points acts as the first layer of location verification. From there, calculations using RSSI values revise the distance between each device and the access points in the area (more details were provided in Chapter 3 Section 3.4.2). Finally, these measurements are compared to determine if both devices are in the same physical location. Based on this outcome, the authentication entity either allows or denies access to requested resources. Two access points (AP1 and AP3) were placed at the two corners of the room, while the third one (AP2) was placed at the center of the wall that is adjacent to the first two access points. This is shown in Figure 4-2 below. The exact positions of the three access points were measured and located manually on the study site as a preliminary step. Subsequently, the user's device was positioned in the room at seven different arbitrary locations. The purpose of choosing seven user device positions is because it allowed for the majority of the specific testing room to be covered with a reasonable range. Choosing more than seven device positions would not yield much, if any, extra coverage or benefit because this amount is the most optimal since it adequately covers the entire room. On the other hand, choosing less than seven appeared

to lower the success rate of the system, in effect leaving certain locations within the room as possible dead zones.

For the experiment, the user's mobile device was located at each of the seven chosen positions. It then scanned the three Wi-Fi access points, along with their corresponding RSSIs. The collected data was used to estimate the user's device location using Equations 5-1 through 5-3, which represent the trilateration system in 2D (a mathematical technique that locates an object on a coordinate plane using three known points).

$$(x - x_1)^2 + (y - y_1)^2 = d_1^2 \quad (\text{Eq. 5-1})$$

$$(x - x_2)^2 + (y - y_2)^2 = d_2^2 \quad (\text{Eq. 5-2})$$

$$(x - x_3)^2 + (y - y_3)^2 = d_3^2 \quad (\text{Eq. 5-3})$$

where:

x and y are the unknown location of an object on a coordinate plane (in Experiment 1, the user device).

x_1 and y_1 are the 2D coordinates of AP_1 .

x_2 and y_2 are the 2D coordinates of AP_2 .

x_3 and y_3 are the 2D coordinates of AP_3 .

d_1 is the distance between AP_1 and the user device.

d_2 is the distance between AP_2 and the user device.

d_3 is the distance between AP_3 and the user device.

Equation 5-4, which is the simple distance equation (see below) that is utilized to calculate a distance between two points on a coordinate plane.

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (\text{Eq. 5-4})$$

where:

x_1 and y_1 are the 2D coordinates of the user's mobile device.

x_2 and y_2 are the 2D coordinates of the user's login machine.

Three different tests (using the first RSSI reading, the average of thirty RSSI readings, and the median of thirty RSSI reading) were performed to test the location accuracy using RSSI as the only feature within the beacon frame. Figure 5-2 shows the access point setup with one result to better illustrate Experiment 1.

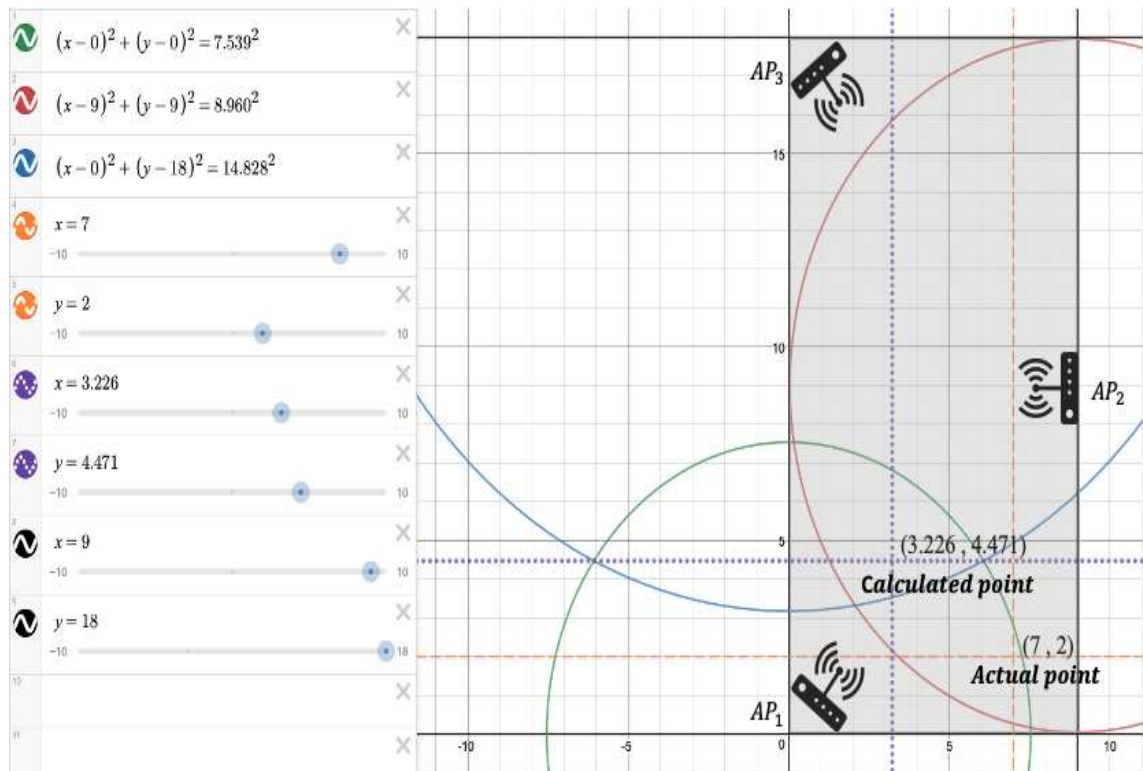


Figure 5-2: Illustration of the access point setup

5.1.1 Calculating the User Device Location Using a Single RSSI Reading

Wi-Fi adapters periodically scan for nearby Wi-Fi access points; therefore, many RSSI readings are obtained over time. For this part of the experiment, the number of

RSSI readings was restricted to a single reading (i.e., only the first RSSI reading was used to calculate the location of the user device). This was done to determine if a single reading (and therefore a single RSSI value) was enough to adequately calculate the location of the user device. The results are shown in Table 5-1, which shows a comparison between the actual location of the device and the calculated location using a single RSSI reading.

Table 5-1: The calculated location using a single RSSI reading

Actual points (x, y)	Calculated points (x, y)	Difference in feet
(2, 17)	(-5.812, 16.635)	7.812
(3, 2)	(6.228, -0.964)	4.382
(4.5, 9)	(0, 9)	4.5
(5, 14)	(4.013, 14.980)	2.216
(6, 2)	(6.083, 4.866)	2.868
(7, 2)	(4.737, 0.120)	2.942
(8, 16)	(9.042, 19.192)	3.358

As can be seen from the table, there are two negative coordinate values.

Obviously the room contains only positive coordinates; however, when calculating points, it is possible that negative values exist which effectively put them outside the room. This occurs because the RSSI readings are sometimes inconsistent. From the results of Table 5-1, it was decided from this experiment that using one RSSI reading is not as beneficial as using more than one (which is based on an average of multiple readings of RSSI values).

5.1.2 Calculating Location Using the Average of Thirty RSSI Readings

For this part of the experiment, the average of thirty RSSI readings was utilized to calculate the location of the user device. This was done to determine if an average of

many RSSI readings provided more accuracy with respect to the location of the user device. The amount of readings (thirty) was selected because additional readings did not factor in significantly. The results are shown in Table 5-2, which shows a comparison between the actual location of the device and the calculated location using an average of thirty RSSI readings.

Table 5-2: The calculated location using the average of thirty RSSI readings

Actual points (x, y)	Calculated points (x, y)	Difference in feet
(2, 17)	(5.173, 15.424)	3.543
(3, 2)	(1.443, 7.478)	5.695
(4.5, 9)	(3.893, 9.869)	1.060
(5, 14)	(4.007, 12.636)	1.687
(6, 2)	(6.968, 2.361)	1.033
(7, 2)	(3.226, 4.471)	4.511
(8, 16)	(5.173, 15.424)	3.543

As can be seen from the results of Table 5-2, using the average reading of RSSI values decreased the difference between the actual points and the calculated points.

5.1.3 Calculating Location Using the Median of Thirty RSSI Readings

For this part of the experiment, we used the median of the thirty RSSI readings to calculate the location of the user device. This was done to determine if the median of the RSSI readings improved the accuracy of the experiment. The results are shown in Table 5-3, which shows a comparison between the actual location of the device and the calculated location using a median of RSSI readings.

Table 5-3: The calculated location using the median of thirty RSSI readings

Actual points (x, y)	Calculated points (x, y)	Difference in feet
(2, 17)	(5.591, 15.842)	3.773
(3, 2)	(1.484, 7.516)	5.721
(4.5, 9)	(3.893, 9.869)	1.06
(5, 14)	(3.957, 12.612)	1.737
(6, 2)	(6.827, 2.606)	1.025
(7, 2)	(4.332, 3.438)	3.031
(8, 16)	(5.392, 10.249)	6.315

As a result, from this experiment, it shows that using a single RSSI reading is quick in terms of time. However, this might not be ideal in terms of accuracy for measuring distances between a user's devices and access points. For example, when the actual point (2,17) was calculated using the first reading of the RSSI the difference between the calculated point and the actual point was 7.812 feet, as can be seen in Table 5-1. However, when the difference of the first actual point (2,17) was estimated through the use of the average, Table 5-2, the resulting difference was 3.543 feet. For estimating the first actual point (2,17) using the median, Table 5-3, the resulting difference was 3.773 feet. As can be seen in Table 5-2 and Table 5-3, the distances between the difference between the actual points and the calculated points decreased. Therefore, using the multiple reading of RSSI values is better than using one reading to estimate. In addition, RSSI values alone do not provide enough information when trying to determine a user's location. Because the proposed system essentially uses the Wi-Fi footprint, more characteristics are needed. Chapter 3 Section 3.3.5 discusses this; specifically, using RSSI, SSID and BSSID values. This will be discussed further in the next section.

5.2 Experiment 2: Testing the Authentication Success Rate

The aim of this experiment was to determine a user's location using the beacon frame characteristics (e.g., SSID, BSSID, RSSI, timestamp, and frequency) – which was discussed in Chapter 3 Section 3.3.5. In this experiment, we evaluated the authentication success rate of the proposed system using RapidMiner because it provides analytics based on real life data [68]. RapidMiner is a software platform that is used as a data science tool and allows for predictive analysis, text mining, and other general data mining use cases. Further modifications were made to the relational database on the authentication entity to create an additional table for the purpose of collecting data from the user login attempts that were made in this experiment, which is shown below in Figure 5-3.

user_id	scan_time	bssid	ssid	level	frequency
41	2020-05-20 13:15:03	50:c7:bf:22:5e:8a	MegaProcTestWifi	-46.000	2462
41	2020-05-20 13:15:03	68:7f:74:86:b5:e8	linksys	-59.000	2437
41	2020-05-20 13:15:03	48:f8:b3:35:82:6d	AP2	-41.000	2437
41	2020-05-20 13:15:03	a0:04:60:75:d8:7e	CyberRoom	-38.000	2437
41	2020-05-20 13:15:03	04:bd:88:df:b2:a4	ENGR122	-77.000	2412
41	2020-05-20 13:15:03	04:bd:88:df:b2:a3	eduroam	-77.000	2412
41	2020-05-20 13:15:03	04:bd:88:df:b2:a2	argjbex	-77.000	2412
41	2020-05-20 13:15:03	04:bd:88:df:b2:a1	LaTechOpenAir	-77.000	2412
41	2020-05-20 13:15:03	04:bd:88:df:b2:a0	LaTechWPA2	-77.000	2412
41	2020-05-20 13:15:03	04:bd:88:df:b2:b4	ENGR122	-81.000	5220
41	2020-05-20 13:15:03	04:bd:88:df:b2:b3	eduroam	-80.000	5220
41	2020-05-20 13:15:03	04:bd:88:df:b2:b2	argjbex	-80.000	5220
41	2020-05-20 13:15:03	04:bd:88:df:b2:b1	LaTechOpenAir	-81.000	5220
41	2020-05-20 13:15:03	04:bd:88:df:b2:b0	LaTechWPA2	-81.000	5220
41	2020-05-20 13:15:03	04:bd:88:df:73:b4	ENGR122	-82.000	5785
41	2020-05-20 13:15:03	04:bd:88:df:73:b3	eduroam	-82.000	5785
41	2020-05-20 13:15:03	04:bd:88:df:73:b2	argjbex	-83.000	5785
41	2020-05-20 13:15:03	04:bd:88:df:73:b1	LaTechOpenAir	-83.000	5785
41	2020-05-20 13:15:03	04:bd:88:df:73:b0	LaTechWPA2	-83.000	5785
41	2020-05-20 13:15:03	9c:1c:12:07:cb:54	ENGR122	-79.000	5745
41	2020-05-20 13:15:03	9c:1c:12:07:cb:53	eduroam	-78.000	5745
41	2020-05-20 13:15:03	9c:1c:12:07:cb:52	argjbex	-79.000	5745
41	2020-05-20 13:15:03	9c:1c:12:07:cb:51	LaTechOpenAir	-78.000	5745
41	2020-05-20 13:15:03	9c:1c:12:07:cb:50	LaTechWPA2	-79.000	5745
41	2020-05-20 13:15:03	00:21:29:03:20:b0	GPULAB	-52.000	5745

Figure 5-3: Data collected before it is run through RapidMiner

In Figure 5-3, the *user-id* column represents the unique identifier of a user. The *scan-time* column represents the timestamp of when this record was inserted. The *bssid* column represents the BSSID of each access point in the area. The *ssid* column represents the name of every access point that was scanned in the area. The *level* column represents the RSSI value that has been measured from the user’s device antenna. The *frequency* column represents the radio frequency type of a scanned access point.

The table was subsequently utilized by RapidMiner to evaluate the authentication success rate. Note that this table was only used during the experiments and was subsequently removed. The system was examined using a number of predictive data modeling techniques to examine the proposed system’s outcomes. The results of four types of predictive data modeling were presented and chosen out of ten models – all of the possible models that fit our experiment – according to the best result, two average results, and the worst result. The chosen models included random forest (which provided the best result), *k*-nearest neighbors (which provided an average result), decision tree (which provided an average result), and random tree model (which provided the worst result). Each predictive data model type and its results are shown separately in the following subsections. In all of the previous subsections, the proposed system’s authentication success rate and accuracy was calculated using Equation 5-5:

$$\frac{TP+TN}{N} * 100 \quad (\text{Eq. 5-5})$$

where:

TP is the true positive, or that access was granted to a legitimate user and both user devices are co-located.

TN is the true negative, or that access was granted to a (potentially) malicious user and both devices are co-located.

N is the total number of access attempts in the experiment.

5.2.1 Random Forest Model

Random forest tree is a supervised learning algorithm. Supervised learning takes an algorithm that has a dataset containing training samples and target samples, and learns the relation between the training samples and their target attributes [69]. The learned relation is the applied to classify new samples without targets [65]. To illustrate how a supervised method works, an example of predicting the salary of an employee based on the number of hours worked which is shown below:

Suppose that Equation 5-6 calculates the salary of the employee:

$$Y = f(x) \quad (\text{Eq. 5-6})$$

where:

x is a function that represents the relation between the salary and number of hours that the employee worked.

x is the input – the number of hours the employee worked.

Y is the output – the salary that the employee is received.

The main goal of the supervised learning algorithm is to predict Y with maximum accuracy for a new input x .

The random forest model is a classification algorithm that uses bagging and randomness in building each decision tree (i.e., a structure in which a node represents a test, each branch represents the output of a test, and each leaf node represents a class label) to create an uncorrelated forest of trees, which is considered more accurate than

any individual tree. The idea of the bagging method is to increase the overall result with a combination of learning methods.

Table 5-4 illustrates the output from the random forest model in a confusion matrix (i.e., a table that describes the performance of an algorithm or an information system). From Table 5-4, the total number of access attempts is 2,671: the number of true positives are 2,296, the number of true negatives are 212, the number of false positives are 97, and the number of false negatives are 66. The prediction performance of the model has an accuracy of 93.90% which was calculated using Equation 5-7, as seen below:

$$\frac{2,296 + 212}{2,671} \times 100 = 93.90\% \quad (\text{Eq. 5-7})$$

Table 5-4: Random forest model result

N = 2,671		Actual	
		Positive	Negative
Predicted	Positive	2,296	97
	Negative	66	212

5.2.2 k-Nearest Neighbors (k-NN) Model

The k-nearest neighbor is a classification and regression algorithm that predicts the target class of a labeled dataset without prior knowledge or assumptions about the data distribution. It applies feature similarity metrics, such as the Euclidean and Manhattan distances, to measure how closely related the features of one class target are to the opposite class target, and to detect the boundary between the two classes.

The Euclidean distance is the distance between two points P and Q, and is defined as the length of PQ . Figure 5-4 illustrates this on a 2D coordinate plane.

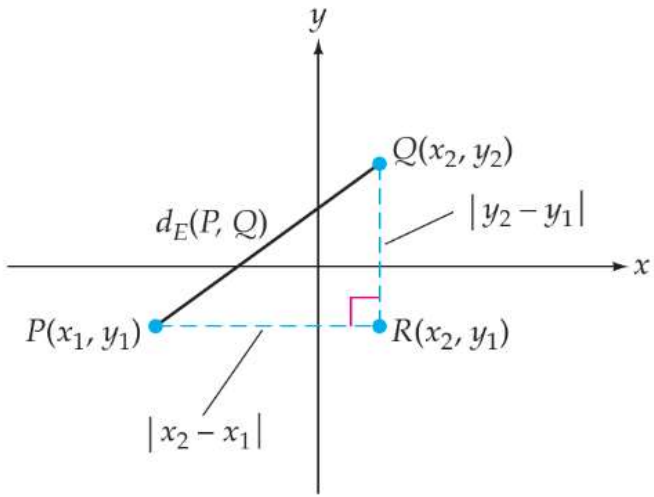


Figure 5-4: Euclidean distance [70]

To better understand Manhattan distance a geometric model of a city is considered, as shown in Figure 5-5.

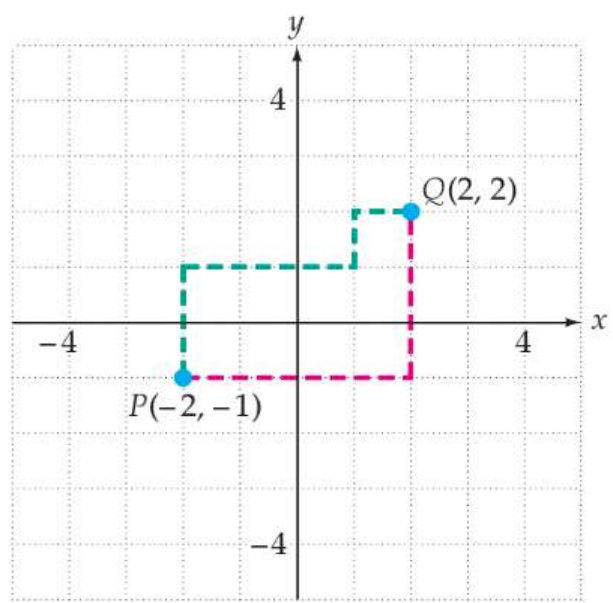


Figure 5-5: Manhattan distance [70]

In this city, the travel direction of all the streets are either straight north and south or straight east and west. One block is the distance between neighboring north-south streets and one block is also the distance between neighboring east-west streets. It is not possible for the travel direction between P and Q to be in a straight path, it has to be along the streets using a short direction. The distance between $P(x_1, y_1)$ and $Q(x_2, y_2)$ are defined in Equation 5-8.

$$d_E(P, Q) = |x_2 - x_1| + |y_2 - y_1| \quad (\text{Eq. 5-8})$$

In this model, a k -value is set to select the top k points or observations that fall within the detected boundary. This process is iterated for all observations in the training set, and each point is assigned to the class target where it is most frequently detected.

Table 5-5 illustrates the confusion matrix of the k -NN model. From Table 5-5, the total number of access attempts are 2,712: the number of true positives are 2,317, the number true negatives are 203, the number of false positives are 108, the number of false negatives are 84. The prediction performance of the model has an accuracy of 92.92% (calculated using Equation 5-3) as seen below:

$$\frac{2,317 + 203}{2,712} \times 100 = 92.92\%$$

Table 5-5: k -nearest neighbors (k -NN) model result

N = 2,712		Actual	
		Positive	Negative
Predicted	Positive	2,317	108
	Negative	84	203

5.2.3 Decision Tree Model

The decision tree is a supervised learning model that learns from a training dataset to build simple decision rules for predicting the target class. The decision tree model can be used to predict the categorical targets of a “Login” target class with the values “Accept” or “Reject.” It can also be applied to predict continuous target classes such as a “Price” target class with the values “less than \$50” and “greater than or equal to \$50.” The model works by creating a tree, and then splits/separates the data at each node using different tree algorithms based on the class type (e.g., categorical or continuous data types: categorical data contains a finite number of distinct groups, while continuous data contains an infinite number of values between any two values which are numeric variables). This split-and-select-features process is then evaluated against different performance measures such as entropy and information gain.

Entropy is a measure of uncertainty of a random variable that depends on splitting what the decision tree carries [71]. Entropy can be utilized to calculate the homogeneity of a sample (i.e., all the variables in a sample have identical traits) in a dataset. Entropy is determined using Equation 5-9.

$$E(D) = -\sum_{i=1}^k P(L_i) \times \log_2(P(L_i)) \quad (\text{Eq. 5-9})$$

Where:

E is the Entropy.

D is the dataset.

$P(L_i)$ is the number ratio of data points in class L_i to the total number of elements in D .

Information gain is the major measure used by decision tree algorithms. The branch with zero is called a leaf node, and if the information gain is greater than zero, it requires further splitting. The information gain can be computed by Equation 5-10.

$$IG(A, D) = E(D) - \sum_{i=1}^k P(T_i) E(T_i) \quad (\text{Eq. 5-10})$$

Where:

E is the Entropy.

D is the dataset.

A is the variable chosen for splitting the dataset.

T_i is the subsets created from D after splitting with A .

$P(T)$ is the ratio of the number of elements in T_i to the number of elements in D .

In this work, a decision tree model was applied with the information gain evaluation method to learn the “Login” class from our data. The output is shown as a confusion matrix in Table 5-6. From Table 5-6, the total number of access attempts are 2,712: the number of true positives are 2,285, the number of true negatives are 214, the number of false positives are 97, the number of false negatives are 116. The prediction performance of the model has an accuracy of 92.15% (calculated using Equation 5-3) as following:

$$\frac{2,285 + 214}{2,712} \times 100 = 92.15\%$$

Table 5-6: Decision tree model result

		Actual	
		Positive	Negative
Predicted	Positive	2,285	97
	Negative	116	214

5.2.4 Random Tree Model

The random tree model is an earlier variant of a tree structure model for classifying a target class from a set of label observations. Random tree structures are generated with two estimators to resolve the high and low variance expected in the data. This model iterates through a subset of the data to obtain the optimal values for the high and low estimators. The estimator parameters are evaluated at a specified confidence level before being passed on for final model construction. In our work, we show the output from running the random tree model on our data in Table 5-7. From Table 5-7, the total number of access attempts are 2,712: the number of true positives are 2,271, the number of true negatives are 167, the number of false positives are 144, and the number of false negatives are 130. Based on the results, the accuracy is 89.90% (calculated using Equation 5-3) as following:

$$\frac{2,271 + 167}{2,712} \times 100 = 89.90\%$$

Table 5-7: Random tree model result

N = 2,712		Actual	
		Positive	Negative
Predicted	Positive	2,271	144
	Negative	130	167

In the first experiment, RSSI readings are utilized to determine a user's location. However, that experiment was conducted using a single RSSI reading – which is quick in terms of time. However, this might not be ideal in terms of accuracy for measuring

distances between a user's devices and access points. In addition, RSSI values alone are not enough to depend on when determining a user's location.

In this experiment, RSSI, SSID, and BSSID readings were utilized to determine a user's location. The Random Forest Model provided the best result (93.90%), the k -NN Model and the Decision Tree Model provided average results (92.92% and 92.15% respectively), and the Random Tree Model provided the worst result (89.90%). However, using more of the beacon frame's characteristics will not increase the accuracy. This experiment utilized characteristics of the beacon frame (e.g., last beacon and timestamp), but those did not prove more accurate in determining a device's location. The RSSI, SSID, and BSSID readings are used together because the RSSI readings are not consistent as determined in experiment 1. Regarding the SSID, there is a possibility that two access points contain the same SSID. Therefore, the BSSID is included because it adds a unique identifier of an access point. Although it is possible that two access points have the same BSSID (e.g., through spoofing of the MAC address), this is very rare.

5.3 Experiment 3: Testing the RSSI Behavior Using Different Phone Models

The aim of this experiment was to test the RSSI behavior over time using different phone models to reasonably determine if mobile devices used by the general public can be reliably used to determine their location using RSSI. In this experiment, an application was developed for installation on the mobile devices. This application scans for the beacon frames of nearby access points and collects relevant data (such as RSSI) that is ultimately packaged and sent to the authentication entity. To thoroughly test this application, it was installed on four separate mobile devices (which included a variety of different models to specifically examine the RSSI behavior using the different models):

Samsung Galaxy S9, Samsung Google OnePlus 3, Samsung Galaxy S4, and Motorola Moto Z. These mobile devices were placed next to each other, and each was set to collect 1,000 RSSI values (a number obtained arbitrarily, but large enough to allow testing of the RSSI behavior over time). Table 5-8 shows the results.

Table 5-8: RSSI measurements

Phone Model	Minimum (dBm)	Maximum (dBm)	Average (dBm)	Standard Deviation
Samsung Galaxy S9	-58	-45	-50.07	2.19
Google OnePlus 3	-59	-48	-50.97	2.14
Samsung Galaxy S4	-60	-50	-52.31	1.75
Motorola Moto Z	-57	-49	-51.74	2.00
All Models	-60	-45	-51.27	2.19

From the table, it can be observed that the RSSI measurements are approximately the same across the four device models. This is likely due to the fact that the internal omni-directional antennas of mobile phones are necessarily small. Thus, the gain value of small omni-directional antennas is theoretically limited [72]. In practice, this does not exceed five dB which results in very little changes in received signal strength. The Federal Communications Commission (FCC) has limits on the gain of Wi-Fi antennas [73], and these limits effectively assure similar RSSI readings across typical mobile devices used by typical users in the proposed system.

5.4 Experiment 4: Testing the Computing and Communication Cost

The aim of this experiment was to calculate the time interval between the user completing a login attempt and the authentication entity delivering an authentication decision. This is important because the proposed system needs to be convenient for users; therefore, the time duration from a login attempt to a login decision must be reasonable

for users to use the system. In this experiment, a test user logged in 50 times, a number arbitrarily selected to provide enough data points. The duration of time that it took the user to login was measured. The output of the elapsed time of each attempt was logged.

Table 5-9 shows the experiment's result.

Table 5-9: Duration of time for 50 login attempts in seconds

Average (seconds)	Minimum (seconds)	Maximum (seconds)	Standard Deviation
8.77	5.98	16.29	1.96

As can be seen in Table 5-10 and Figure 5-6, the proposed system achieved better results when compared to a number of competing 2FA systems (SMS/Voice message, Google Authenticator, USB security key, and Google prompt/push notification) that were used to login into a Gmail account. These methods had a login duration range of 14 to 590 seconds [74], as seen in Figure 5-7.

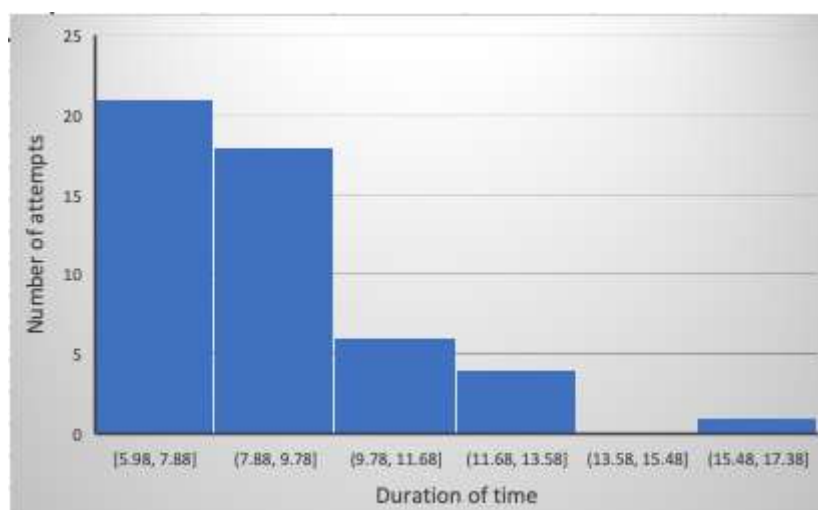


Figure 5-6: Total time of logging in (in seconds) for the proposed system

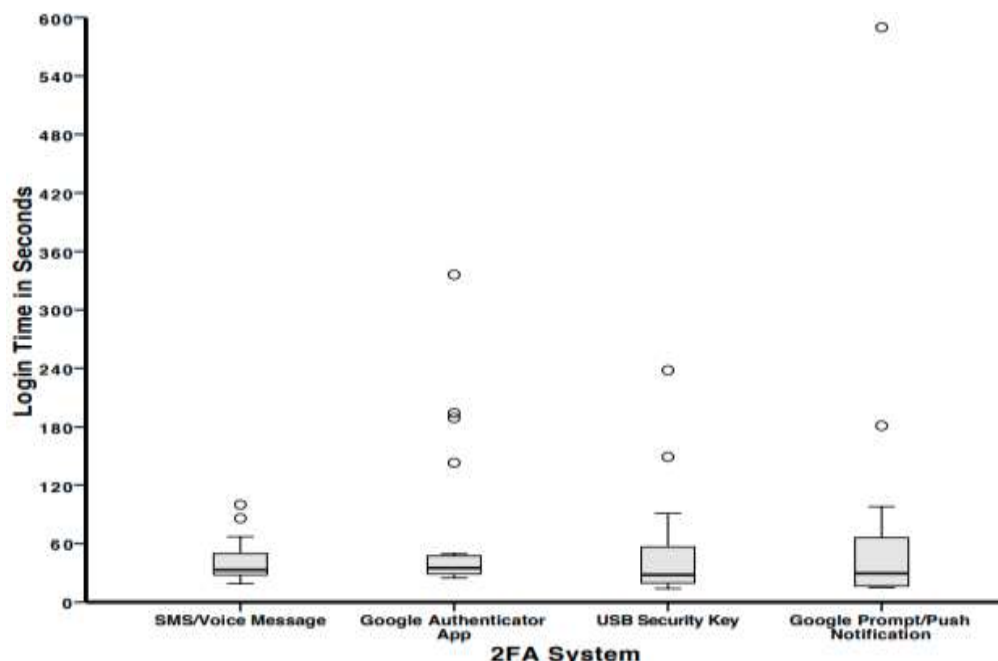


Figure 5-7: Total time of logging in (in seconds) for different types of 2FA systems [74]

5.5 Experiment 5: Testing the Proposed System's Robustness Through Implementation

The aim of this experiment was to test the robustness of the proposed system through implementation of the complete verification processes (more details were provided in Chapter 3 Section 3.3.2) with a minimum number of access points. The rate of Wi-Fi devices scanning for beacon frames was set to one minute because most devices support scanning at 30 second intervals or more. Scanning more frequently results in duplicate RSSI values from previous scans. In the authentication entity, the acceptable range of a user's device to Wi-Fi access points was set to 10-meters (which is an area that can be used to simulate a typical office environment). Furthermore, the number of required nearby access points in the Wi-Fi footprint was set to one or more in order to test the robustness of the proposed system by implementing the verification processes with the minimum number of access points. This experiment was attempted twenty times

(i.e., this number was selected because every time the experiment was run the same result was obtained) and Figure 5-8 shows one of the results.

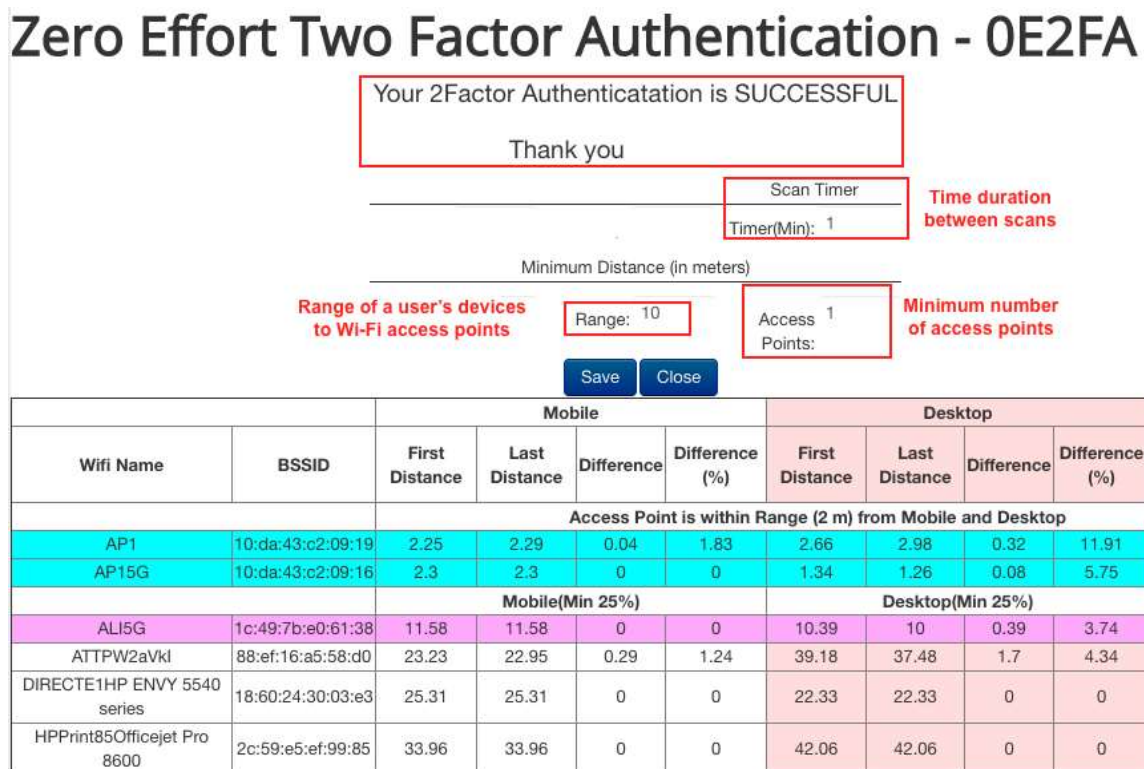


Figure 5-8: Result of testing the proposed system's robustness through implementing the verification processes

Figure 5-8 shows a user's successful login attempt. The information that is beneath the *Wifi Name* column represents the SSID of all the access points in the area. The *BSSID* column represents the BSSID of all the access points in the area.

Also, in the mentioned figure (Figure 5-8) the data that is beneath the *Mobile* column is the data that is gained from the mobile device. The *First Distance* column represents the first calculated distance between the mobile device and an access point in meters when the user first logs in. The *Last Distance* column represents the last calculated distance between the mobile device and an access point in meters of the last

scan. The *Difference* column represents the difference between the *First Distance* and *Last Distance* in meters. The *Difference (%)* column represents the difference between the *First Distance* and *Last Distance* as a percentage.

Also, in Figure 5-8 the data that is beneath the *Desktop* column is the data gained from the login machine. The *First Distance* column represents the first calculated distance between the login machine and an access point in meters when the user first logs in. The *Last Distance* column represents the last calculated distance between the login machine and an access point in meters at the time of scan. The *Difference* column represents the difference between the *First Distance* and *Last Distance* in meters. The *Difference (%)* column represents the difference between the *First Distance* and *Last Distance* as a percentage.

As can be observed from Figure 5-8, there are two access points that meet all of the parameters (denoted in blue). There is one access point that meets all of the requirements except for being within range of ten meters (denoted in purple); therefore, the proposed system ignored it.

As a result, the proposed system successfully implemented the verification processes with two access points. Furthermore, all the authentication attempts from this experiment shows that the proposed system has the ability to implement the verification processes with a minimum number of access points.

5.6 Experiment 6: Lack of Access Points

The aim of this experiment was to test the robustness of the proposed system in implementing the access policies stated earlier: (1) There exists at least a predefined number of overlapping access points detected by both devices; and (2) The estimated

distance from the devices to all access points falls within a predefined range. Specifically, this experiment ensures this with a single access point. In the worst case, a user may have only one access point visible in the surrounding area. For this test, the Wi-Fi footprint was fixed at one access point in order to further examine the robustness of the proposed system. The acceptable range of a user's device to the Wi-Fi access point was set to two meters to test the accuracy and robustness of the proposed system in such a small range from the access point. The proposed system was set to authenticate at intervals of one minute as in the last experiment. This was repeated for a total length of 20 minutes, this length of time was selected because the result would begin to repeat after this amount of time (i.e., the devices only have a certain scan range and after this amount of time the device will get similar values since the change in distance in between scans is negligible). The threshold set for the range between the user's two devices was set to three meters (i.e., they must be no more than three meters to be considered co-located) to examine the robustness of the proposed system with a lack of access points in a small environment. The results in Table 5-10 show the estimated distance between the two devices and the access point.

Table 5-10: Result using one access point

	Mobile Device			Login Machine		
	Minimum	Maximum	Average	Minimum	Maximum	Average
Calculated Distance	1.7 meter	2.0 meters	1.8 meter	1.8 meter	2.5 meters	2.0 meters
Percent Error (from 2m)	15%	0%	10%	10%	25%	0%

As seen in Table 5-10, the difference between the actual distance of two meters and the calculated distance is less than or equal to half a meter for both devices.

Therefore, with only one access point the proposed system was able to implement and authenticate the user.

5.7 Experiment 7: Mobile Device Application Security

The goal of this experiment was to test the security of the mobile device application against malicious activity. For example, suppose that an attacker attempts to login to the authentication entity through the mobile device application using credential information that is not registered under the compromised user's mobile device (i.e., the mobile device's IMEI and device ID are not registered to the user that the attacker is attempting to login as). In this experiment, the proposed system's mobile device application was measured against a login request with credential information that is not associated with the mobile device IMEI and device ID. For this experiment, five logins (i.e., 20 logins were attempted, but following the first five attempts all the results were the same) were attempted with a number of usernames and passwords that were not associated with the mobile device IMEI of a specific user account. As can be seen below in Figure 5-9, an example login was simulated, and the application rejected the request.



Figure 5-9: Rejection request

As a result of the experiment, associating the mobile device application with the unique identifiers of a mobile device (i.e., IMEI, device ID) was useful in preventing an unauthorized user from logging in (more details about this type of attacks were provided in Chapter 5 Section 5.1.2).

5.8 Experiment 8: Battery Consumption

The goal of this experiment was to determine how much of the battery is being consumed by the mobile device application in its required tasks in the proposed system. As mentioned in Chapter 3, Section 3.4.1, the mobile device application runs in the background of the mobile device. In this experiment, a Samsung S6 was utilized and was first fully charged (100%). Once the battery percentage reached the 49%, the battery usage by the mobile application was checked. As can be seen in Figure 5-10 the battery usage for the mobile application (02FAuth) is less than one percent.



Figure 5-10: Battery consumption

From this, we conclude that the mobile device application does not result in significant battery consumption while it runs in the background. However, a user may disable the application by selecting *FORCE STOP* in one of the phone's app setting screens (as seen in Figure 5-11).

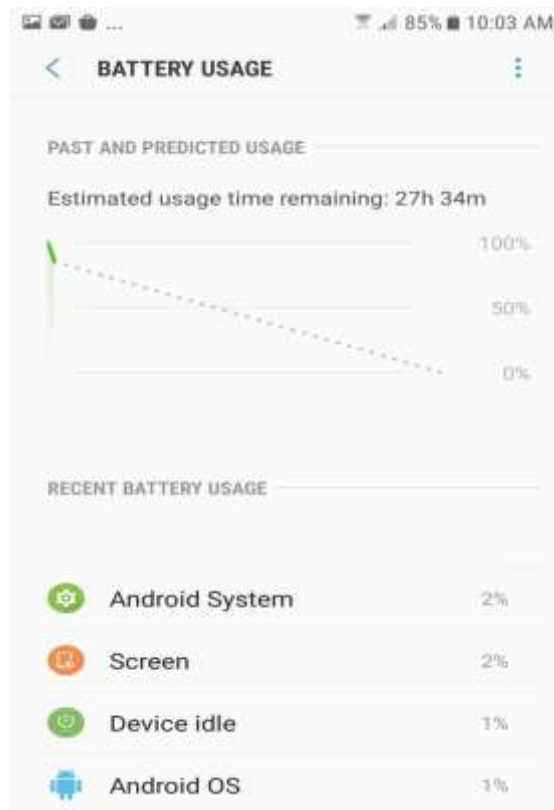


Figure 5-11: Battery consumption

5.9 Conclusion

In this chapter, several experiments were performed. These experiments were used to test and address aspects of the proposed system, and the results were incorporated into design decisions. The location accuracy, the authentication accuracy, the RSSI behavior over different types of mobile devices, the computing and communication costs,

and the robustness of the proposed system were examined under various circumstances.

In the next chapter, this work is summarized and possible future directions are discussed.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

Today, digital information decisively permeates all aspects of modern society [75]. In such a connected world, one of the critical factors in maintaining the security of transmitted data is user authentication [76] [77]. User authentication is considered to be a critical factor in the first layer of security, which establishes confidence in the user identities presented to an information system [78]. Over time, the definition of user authentication has not changed; however, a simple password is no longer the only factor used for authenticating a user [79]. 2FA was proposed to provide a high level of security by adding more than one layer of authentication. For the most part, these layers of authentication are based on knowledge, ownership, biometrics, and/or the behaviors of users [80].

At present, there are many 2FA systems in use where security requirements are higher than usual [81] [82]. If we categorize them based on the submitted effort of users, then a few fall into the category of zero effort. One of the strengths of the proposed system is that it is a zero-effort two-factor authentication system. The proposed system is a 2FA security scheme that manages access to resources with no required effort from a user by utilizing what is in the user's environment. This research proposes a 2FA system

that attempts to take advantage of the current IEEE 802.11 infrastructure, making the proposed system directly implementable within existing infrastructure and widely usable.

Currently, there is relatively little research that addresses 2FA in a zero-effort way for users. The existing zero-effort studies largely depend on the ambient sound in the user's environment. However, these studies may not fit all environments because they depend on the ability to listen to ambient sound. In addition, not all devices are equipped with a built-in microphone. Also, the user may be unable to find an environment with the right noise levels for the software when needed [44]. If there is a single loud source of noise, such as a television playing in the background, construction, or music, it may also be possible to trick the system into perceiving the mobile device to be in the same location as the authentication device [57]. The possibility of "overhearing" important, private information can lead to significant security vulnerabilities or infringe on the privacy of users [58]. Most of the current work results in more involvement by users through required effort in implementing the second layer of authentication. This could be considered inconvenient for the user.

The proposed solution aims to add an extra layer of authentication by finding something unique that is in a user's *environment*; specifically, by utilizing information at ambient access points (a Wi-Fi *footprint*). This system depends on something that a user *knows*, something that a user *owns*, and something that is in the user's *environment* – arguably the most significant contribution of this work. Primarily due to the proposed system removing the "weak link" from the second authentication factor, the proposed system facilitates the use of 2FA in more systems.

6.2 Future Work

The goal of this work is to present a 2FA system that requires no effort from users when implementing the second layer of authentication. Although we consider this goal to have been achieved, there are many improvements that could be made in order to extend the work. Below are some potential future directions:

6.2.1 Collecting Wi-Fi Footprints Through the Login Machine

In the proposed system, a custom application serves as the gateway to accessing resources through the authentication entity. With the application, a user's devices can automatically collect beacon frames from visible Wi-Fi access points and select specific features for use by the authentication entity for the purposes of granting access to resources. In future work, exploring a way to remove the need for a custom application on the user's login device and utilize a web browser instead to scan and collect data from the beacon frames would be ideal. Since users enter their username and password using this mechanism, additionally implementing the login machine's scanning of the Wi-Fi footprint would be ideal and remove the need for installing the application on the login machine.

6.2.2 Continuous Authentication

In Sections 5.3.1 and 5.3.2, some potential issues that a user might face while using our system were discussed. The prospective solutions for these issues can be implemented with continuous authentication and still require zero effort from the user. In future work, this continuous authentication feature could be considered as one of the primary functions in the proposed system to increase its security without placing any additional burden on users. Once the user has installed the applications, the proposed

system requires no more intervention other than simply entering a username and password.

6.2.3 Estimating the Distance Between the User's Devices

As explained in Chapter 3, the proposed system implements a second layer of authentication that utilizes something unique in a user's environment. Every access point broadcasts messages that carry distinctive information that can be used to identify a user's environment. This information can subsequently be used to influence the outcome of user authentication to requested resources. In this study, data within these broadcast messages were utilized to implement a second layer of authentication by determining if two devices are in the same physical location. In future work, a method of more precisely calculating the distance between the user's two devices could be studied in order to increase the accuracy of the system. This would be particularly relevant to the continuous authentication extension discussed above.

APPENDIX A

SOURCE CODE

This appendix contains the source code that is implemented in the proposed system. The source code is written in Python. The first listing includes functions implemented at the authentication entity such as: (1) receiving the collected data (i.e., SSID, BSSID, RSSI readings) from a user's devices; (2) calculating the distances between a user's devices (i.e., mobile device and login machine) with every access point in an area; (3) accessing the proposed system's database; and (4) inserting calculated distances and collected data (i.e., SSID, BSSID, RSSI readings) into the proposed system's database.

(1)

```

#!/usr/bin/python3
import sys
import os
import math
import mysql.connector
from mysql.connector import Error
from datetime import datetime
import time

tables = ["mobile","desktop"]
# the equation that is used to calculate the distance between a user's
devices and every access point//

def getDist(rssi, frq, DevID): # (RSSI, Wave Le, Device ID) //
    try:
        PLE = [2.7, 2.7] # [Mobile, Desktop] Path Loss Exponent value //
        WaveLen = [0.125, 0.05] # [2.4g,5g] Wavelength Values //
        if int(frq) < 5000: w = 0
        else: w = 1
        PL_log = 20.0 - float(rssi)
        PL0 = 20.0 * math.log10((4.0*math.pi)/WaveLen[w])
        d = float(pow(10,(PL_log - (PL0))/(10.0*PLE[DevID])))
    except:
        d = 0.0, 0.0
    finally:
        return d, d;

# Accessing the proposed system's database //

def Select(sql):
    rec = [];
    try:
        conn = mysql.connector.connect(host='localhost', database='cron_jobs',
user='root', password='password')
        if conn.is_connected():
            cur = conn.cursor()
            cur.execute(sql)
            rec = cur.fetchall()
    except:
        rec = [];
    finally:
        if conn.is_connected():
            conn.close()
        return rec;

# Insert Data into the database //

def Upload(devID, rssi, freq, Tot, n, userid, bssid, scantime, firstDist,
isEnd):
    ret = [Tot, n];
    try:

```

```

    Dist, DistF = getDist(rssi,freq,devID); Tot += Dist; n += 1;
    if n == 1 or isEnd:
        saveLog('\tUserID: '+str(userid)+'\t'+tables[devID].capitalize()+': \t
BSSID: '+str(bssid)+'\tTime: '+str(scantime)+'\t RSSI: '+str(rssi)+'\tDist:
'+str(round(firstDist,3))+' --> '+str(round(Tot/n,3))+'\n');
        saveDist(devID, firstDist, (Tot/n), userid, bssid);
        Tot = 0; n = 0;
        sql = 'UPDATE IGNORE '+tables[devID]+'_data SET recorded = 0 WHERE
user_id='+userid+' AND bssid='\'+bssid+'\'' AND scan_time <= \''+scantime+'\'';
        Execute(sql);
        if devID == 0:
            dm_sql = 'INSERT INTO classification_data (mac, m_dist,m_freq,
m_level,mobile_time) values
(\'+bssid+\'','+str(Dist)+','+str(freq)+','+str(rssi)+', \''+scantime+\'');'
            else:
                dm_sql = 'UPDATE IGNORE classification_data SET d_dist='+str(Dist)+',
d_freq='+str(freq)+', d_level ='+str(rssi)+', desktop_time = \''+scantime+\'
WHERE mac = \''+bssid+\' AND desktop_time IS NULL;
            Execute(dm_sql);
            #saveLog(sql)
            ret = [Tot, n];
        except Exception as e:
            PrintErr();
    finally:
        return ret;

```

The second listing includes functions at the authentication entity such as: (1) finding devices that have active sessions; (2) finding the overlapping access points; and (3) creating the overlapping access points table.

(2)

```

#!/usr/bin/python3
from myscript import getDist, saveDist, Select, Execute, saveLog, Upload
import sys
import os
import math
from datetime import datetime
import time

        ##### Find devices that have active sessions
tables = ["mobile","desktop"]
for i in range(N):
    dev_rec = Select('SELECT user_id FROM cron_jobs.device_data WHERE
status=1;');

        ##### Finding the overlapping access points ) ###
m_Tot, m_Ave, d_Tot, d_Ave, Tot = [0, 0, 0, 0, 0]
d_bssid = ''; m_bssid = ''; bssid = '';
for dev_row in dev_rec:
    USER_ID = str(dev_row[0]);
    sql = 'SELECT DISTINCT d.bssid, d.ssid, '\
        'max(d.level), max(d.scan_time) dscan_time,
max(d.frequency), max(d.last_beacon_time), max(d.scan_count), d.recorded, '\
        'max(m.level), max(m.scan_time) mscan_time,
max(m.frequency), max(m.time_last_seen), max(m.scan_count), m.recorded,
IFNULL(ms.desktop_first,-1), IFNULL(ms.mobile_first,-1) '\
        'FROM desktop_data d INNER JOIN mobile_data m ON m.BSSID =
d.BSSID AND m.user_id = d.user_id '\
        'LEFT JOIN measurement_data ms ON (ms.user_id = d.USER_ID
and ms.BSSID = d.BSSID) '\
        'WHERE d.user_id = '+USER_ID+' '\
        'AND (d.recorded = -1 AND m.recorded = -1) '\
        'GROUP BY d.bssid, d.ssid, '\
        'd.recorded, '\
        'm.recorded, '\
        'IFNULL(ms.desktop_first,-1), IFNULL(ms.mobile_first,-1) '\
        'ORDER BY d.bssid, dscan_time, mscan_time;';

    rec = Select(sql);
    l = len(rec)
    if (l > 0): # Minimum matching more than zero
        #saveLog('\n-----\n');
        saveLog(str(l)+' Matching Records Found\n ');
        saveLog('\n-----\n');

saveLog('*****\n');

```

```

# Minimum matching more than zero Creating the overlapping access points table
j, n, d_j, M_n, D_n = [0, 0, 0, 0, 0];
M_Tot, D_Tot = [0.0, 0.0];
Tot += 1
checkDev = True;
while True:
    n += 1
    if j == 0 or BSSID != rec[j][0]:
        checkDev = True;
        BSSID, SSID, D_SCANTIME, M_SCANTIME, D_FIRST, M_FIRST = [rec[j][0],
rec[j][1], rec[j][3], rec[j][9], rec[j][14], rec[j][15]]
        DevID, m_j, d_j, M_n = [0, 0, j, 0]
        sql = 'INSERT IGNORE INTO measurement_data (user_id,
bssid,mobile_diff,desktop_diff,diff) VALUES ('+USER_ID+',
\''+BSSID+\'\',1.0,1.0,1e3);';
        saveLog('\n-----
-----\n['+SSID+']\n');
        Execute(sql);
        isEnd = False;

        if DevID == 0 and D_SCANTIME == rec[j][3]:
            Rssi, ScanTime, Freq, BeaconTime, ScanCount, Proc = rec[j][8:14];
            isEnd = (j+1 == 1 or BSSID != rec[j+1][0] or D_SCANTIME !=
rec[j+1][3]);
            M_Tot, M_n = Upload(DevID, Rssi, Freq, M_Tot, M_n, USER_ID, BSSID,
str(ScanTime), M_FIRST, isEnd);
            m_j += 1; j += 1;

        if isEnd or DevID == 1:
            DevID = 1;
            Rssi, ScanTime, Freq, BeaconTime, ScanCount, Proc = rec[d_j][2:8];
            isEnd = (d_j+m_j+1 >= 1 or BSSID != rec[d_j+m_j][0]);
            D_Tot, D_n = Upload(DevID, Rssi, Freq, D_Tot, D_n, USER_ID, BSSID,
str(ScanTime), D_FIRST, isEnd);
            d_j += m_j; j = d_j;
            if j >= 1:

saveLog('*****\n');
        break;
        sql = 'UPDATE IGNORE measurement_data SET mobile_diff = ABS(mobile_last-
mobile_first)/mobile_first, desktop_diff = ABS(desktop_last-
desktop_first)/desktop_first WHERE mobile_first IS NOT NULL AND desktop_first
IS NOT NULL;';
        Execute(sql);
        sql = 'UPDATE IGNORE measurement_data SET diff = ABS(mobile_last-
desktop_last) WHERE mobile_last IS NOT NULL AND desktop_last IS NOT NULL;';
        Execute(sql);
        rec.clear()
        dev_rec.clear()
        time.sleep(T)

```

```
except Exception as e:  
    exc_type, exc_obj, exc_tb = sys.exc_info()  
    fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]  
    print(exc_type, fname, exc_tb.tb_lineno)  
    saveLog('Error')  
    #print("Unexpected error:", sys.exc_info())
```


REFERENCES

- [1] Ackerman, P., Impediments to adoption of two-factor authentication by home end-users, (2019).
- [2] Burr, W. et al., *Electronic authentication guideline*, (2013).
- [3] Shirey, R., *RFC 4949 - internet security glossary, version 2*. Tools.ietf.org. (2019).
- [4] D. Barrett, R. Byrnes and R. Silverman, *SSH: The secure shell*. 93, Beijing: O'Reilly, (2005).
- [5] S. K. Modi, *Biometrics in identity management concepts to applications*. Boston: Artech House, (2011).
- [6] D. Chiras, *Human biology*. Jones & Bartlett Publishers, (2013).
- [7] M. Lockie and R. Reidy, *The biometric industry report*. Oxford: Elsevier Advanced Technology, (2002).
- [8] Mammeri, Z., Introduction to IEEE 802.11 standards. *Semantic Scholar*. (2018).
- [9] Madlmayr, G., Langer, J., Kantner, C. and Scharinger, J., NFC Devices: Security and privacy. *Third International Conference on Availability, Reliability and Security*, 2008. 642-647.
- [10] IGI Global, *Mobile computing and wireless networks: Concepts, methodologies, tools, and applications*, (2015).
- [11] Banerji, S. and Chowdhury, R. On IEEE 802.11: Wireless lan technology. *International Journal of Mobile Network Communications & Telematics*, 3(4), 45-64, (2013).
- [12] Anon, *Demographics of mobile device ownership and adoption in the United States*. Pew Research Center: Internet, Science & Tech, (2019).

- [13] Gupta, V. and Kumar Rohil, M., Bit-stuffing in 802.11 beacon frame: Embedding non-standard custom information. *International Journal of Computer Applications*, 63(2), 6-12 (2013).
- [14] Gupta, Vishal, et al., Information Embedding in IEEE 802.11 Beacon Frame, *National Conference on Communication Technologies & its Impact on Next Generational Computing CTNGC 2012*, Mohan Nagar, Ghaziabad, (2012).
- [15] Al Shourbaji, I., An overview of wireless local area networks (WLAN), (2013).
- [16] Dorda, P., 2019. *An Introduction to Computer Networks*, Chicago, IL.
- [17] Anon, IEEE Standard for information technology; Telecommunications and information exchange between systems. local and metropolitan area networks—specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Computer Society*, (2007).
- [18] Terry, J. and Heiskala, J., *OFDM wireless LANs: A theoretical and practical guide*, Indianapolis, Indiana: SAMS, (2002).
- [19] Wescott, D., Coleman, D., Mackenzie, P. and Miller, B. *CWAP: Certified wireless analysis professional official study guide: Exam PW0-270*, Indianapolis, IN: Wiley, (2011).
- [20] *The International Council of Electronic Commerce Consultants*, E., Ethical hacking and countermeasures: Secure network operating systems and infrastructures (CEH) book 4 of ethical hacking and countermeasures, EC-Council Press 2nd ed., Boston, MA: Cengage Learning, (2016).
- [21] Coleman, D., Westcott, D., Harkins, B. and Jackman, S., *CWSP Certified wireless security professional official study guide: Exam PW0-204*, Indianapolis, Ind.: Wiley Pub. (2010).
- [22] Held, G., *Securing wireless LANs*, Hoboken, NJ: Wiley, (2003).
- [23] B. Shin, *A practical introduction to enterprise network and security management*. CRC Press, (2017).
- [24] *Standards-oui.ieee.org*, 2020. [Online]. Available: www.standards-oui.ieee.org/oui/oui.txt
- [25] Ciampa, M., *CWNA guide to wireless LANs 3rd ed.*, Boston, Mass.: Course Technology, CENGAGE Learning, (2013).

- [26] AlQahtani, A., Alamleh, H., Gourd, J. and Alnuhait, H., *TS2FA: Trilateration System Two Factor Authentication*. 3rd International Conference on Computer Applications & Information Security (2020).
- [27] AlQahtani, A., Alamleh, H. and Gourd, J., 0EISUA: Zero effort indoor secure user authentication. *IEEE Access*, 8, 79069-79078, (2020).
- [28] A. AlQahtani and J. Gourd., 0E2FAUE: Zero effort two factor authentication based on user's environment. *IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications* (2020).
- [29] A. AlQahtani, H. Alamleh, and J. Gourd., BF2FA: Beacon frame two factor authentication. In the *7th IEEE International Conference on Cyber Security and Cloud Computing* (2020).
- [30] Basavala, S., Kumar, N. and Agarrwal, A., Authentication: An overview, its types and integration with web and mobile applications. *2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing*, (2012).
- [31] Shaju, S. and V, P., BISC authentication algorithm: An efficient new authentication algorithm using three factor authentication for mobile banking. *2016 Online International Conference on Green Engineering and Technologies* (2016).
- [32] Yu, J. and Brune, P., No security by obscurity - why two factor authentication should be based on an open design. *Proceedings of the International Conference on Security and Cryptography*, (2011).
- [33] Al-Sahwan, G. et al., A strong and practical authentication mechanism using PassText and OTP. *2018 21st Saudi Computer Society National Computer Conference*, (2018).
- [34] Uhl, X. and Brezina, C., *A primary source investigation of the Industrial Revolution* 1st ed., The Rosen Publishing Group, Inc.,(2019).
- [35] Fujii, H. and Tsuruoka, Y., SV-2FA: Two-factor user authentication with SMS and voiceprint challenge response. *8th International Conference for Internet Technology and Secured Transactions* (2013).
- [36] Reese, K., *Evaluating the usability of two-factor authentication*. MS. Brigham Young University, (2018).
- [37] Sodhi, B., Using dropped call as an authentication factor. *2015 IEEE International Conference on Computer and Information Technology, Pervasive Intelligence and Computing*, (2015).

- [38] Hunt, V., Puglia, M. and Puglia, A., *A guide to radio frequency identification*, New York: J. Wiley, (2007).
- [39] Mathew, M. and Divya, R., Super secure door lock system for critical zones. *2017 International Conference on Networks & Advances in Computational Technologies*, (2017).
- [40] Liu, H. and Zhang, Y., An improved one-time password authentication scheme. *2013 15th IEEE International Conference on Communication Technology*, (2013).
- [41] Tombeng, M. and Laluyan, H., Prototype of authentication system of motorcycle using RFID implants. *2017 5th International Conference on Cyber and IT Service Management*, (2017).
- [42] Tiwari, S., An introduction to QR code technology. *2016 International Conference on Information Technology*, (2016).
- [43] Rodrigues, B., Chaudhari, A. and More, S., Two factor verification using QR-code: A unique authentication system for Android smartphone users. *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 457-462, (2016).
- [44] Schürmann, D. and Sigg, S., Secure communication based on ambient audio. *IEEE Transactions on Mobile Computing*, 12(2), 358-370, (2013).
- [45] Krombholz, K., Fruhwirt, P., Rieder, T., Kapsalis, I., Ullrich, J. and Weippl, E., 2015. QR code security: How secure and usable apps can protect users against malicious QR codes. *2015 10th International Conference on Availability, Reliability and Security*, (2015).
- [46] Pratama, A. and Prima, E., 2016. 2FMA-NetBank: A proposed two factor and mutual authentication scheme for efficient and secure internet banking. *2016 8th International Conference on Information Technology and Electrical Engineering*, (2016).
- [47] Ahson, S. and Ilyas, M., *Near field communications handbook*, CRC Press, (2016).
- [48] McHugh, S. and Yarmey, K., *Near field communication*, San Rafael, California: Morgan & Claypool Publishers (2014).
- [49] Hufstetler, W., Ramos, M. and Wang, S., NFC unlock: Secure two-factor computer authentication using NFC. *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems*, 507–510, (2017).

- [50] Coskun, V., Ozdenizci, B. and Ok, K., *The survey on near field communication sensors*, 15(6), 13348-13405, (2015).
- [51] ENFROY, A., *27 Best name generators*, Domain, Company, and Random, (2019).
- [52] Chattha, N., NFC - Vulnerabilities and defense. *2014 Conference on Information Assurance and Cyber Security*, 35–38, (2014).
- [53] Chen, C., Lin, I. and Yang, C., NFC attacks analysis and survey. *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 458–462, (2014).
- [54] Luo, J., Tsai, M., Lo, N., Kao, C. and Yang, M., *Ambient audio authentication. Mathematical Biosciences and Engineering*, 16(6), 6562-6586, (2019).
- [55] Mahansaria, D. and Roy, U., Secure authentication for ATM transactions using NFC technology. *2019 International Carnahan Conference on Security Technology*, (2019).
- [56] Zhu, X., Yu, S. and Pei, Q., QuickAuth: Two-factor quick authentication based on ambient sound. *2016 IEEE Global Communications Conference*, (2016).
- [57] Shrestha, B., Shirvanian, M., Shrestha, P. and Saxena, N., The sounds of the phones: Dangers of zero-effort second factor login based on ambient audio, (2016).
- [58] Wang, M., Zhu, W., Yan, S. and Wang, Q., SoundAuth: Secure zero-effort two-factor authentication based on audio signals. *2018 IEEE Conference on Communications and Network Security*, (2018).
- [59] Shrestha, B., Shirvanian, M., Shrestha, P. and Saxena, N., Sound-Proof: Usable two-factor authentication based on ambient sound. *Proceedings of the 2016 ACM*, (2016).
- [60] Syed, I., Kim, B., Roh, B. and Oh, I., A novel contention window backoff algorithm for IEEE 802.11 wireless networks. *2015 IEEE/ACIS 14th International Conference on Computer and Information Science* (2015).
- [61] Singh, M., Adzman, K. and Hassan, R., Near field communication technology security vulnerabilities and countermeasures. *International Journal of Engineering & Technology*, 7(4), 298-305, (2019).
- [62] Mulliner, C., Vulnerability analysis and attacks on NFC-enabled mobile phones. *2009 International Conference on Availability, Reliability and Security*, 695-700, (2009).

- [63] Howes, T., Smith, M. and Good, G. *Understanding and deploying LDAP directory services, second edition*. Addison-Wesley Professional, (2003).
- [64] Kar, A., *Digital nations - smart cities, innovation, and sustainability*. Springer International Publishing, (2017).
- [65] Li, X. and Li, J., *Quality based content delivery over the internet*, Shanghai: Shanghai Jiao Tong Univ. Press, (2011).
- [66] Lai, J. and Wibowo, S., How service convenience influences information system success. *International Journal of Future Computer and Communication*, 217-220, (2012).
- [67] Netgear R6400 AC1750 Smart WiFi Router, (2020).
<https://www.techspot.com/products/routers/netgear-r6400-ac1750-smart-wifi-router.121973/>. Accessed: 2020- 05- 06.
- [68] Key features of rapidminer studio | RapidMiner, (2020).
<https://rapidminer.com/products/studio/feature-list/>.
- [69] Ramasubramanian, K. and Moolayil, J., *Applied supervised learning*, Packt Publishing, Limited, (2019).
- [70] Aufmann, R., Lockwood, J., Nation, R. and Clegg, D. *Mathematical Excursions, Enhanced Edition*, (2014).
- [71] Thomas, T., P. Vijayaraghavan, A. and Emmanuel, S., *Machine learning approaches in cyber security analytics*. Springer Singapore, (2019).
- [72] Chu, L., Physical limitations of omni-directional antennas. *Journal of Applied Physics*, 19(12), 1163- 1175, (1948).
- [73] *FCC rules and regulations 2.4 & 5 GHz bands*. 2018. Air802.com
- [74] Acemyan, C., Kortum, P., Xiong, J. and Wallach, D. 2FA might be secure, but it's not usable: A summative usability assessment of Google's two-factor authentication (2fa) methods. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 62(1), 1141-1145, (2018).
- [75] Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T. and Koucheryavy, Y., Multi-factor authentication: A survey. *cryptography*, 2(1), 1, (2018).
- [76] Roy, S.; Khatwani, C., Cryptanalysis and improvement of ECC Based authentication and key exchanging protocols. *Cryptography*, (2017).

- [77] Alomar, N.; Alsaleh, M.; Alarifi, A., Social authentication applications, attacks, defense strategies and future research directions: A systematic review. *IEEE Commun. Surv. Tutor*, (2017).
- [78] Nicholson, D., *Advances in human factors in cybersecurity*, Springer, (2018).
- [79] T Benarous, L.; Kadri, B.; Bouridane, A., A survey on cyber security evolution and threats: Biometric authentication solutions. In *Biometric Security and Privacy*; Springer: Berlin, Germany, (2017).
- [80] Harini, N.; Padmanabhan, T.; others. 2CAuth: A new two factor authentication scheme using QR-code. *Int. J. Eng. Technol.*, (2013).
- [81] Coventry, L.; De Angeli, A.; Johnson, G., Usability and biometric verification at the ATM interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Ft. Lauderdale, FL, USA, 5–10, April (2003).
- [82] Tahir, H.; Tahir, R., BioFIM: Multifactor authentication for defeating vehicle theft. In *Proceedings of the World Congress on Engineering*, London, UK, (2008).