Spring 5-25-2019

Unobtrusive Location-Based Access Control Utilizing Existing IEEE 802.11 Infrastructure

Hosam Alamleh
Louisiana Tech University

UNOBTRUSIVE LOCATION-BASED ACCESS CONTROL UTILIZING EXISTING IEEE 802.11 INFRASTRUCTURE

by

Hosam Alamleh, M.S

A Dissertation Presented in Partial Fulfillment
of the Requirements of the Degree
PhD of Engineering

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

May 2019

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

APRIL 3, 2019
Date

We hereby recommend that the dissertationprepared under our supervision

Hosam Alamleh, M.S

entitled UNOBTRUSIVE LOCATION-BASED ACCESS CONTROL UTILIZING EXISTING IEEE 802.11 INFRASTRUCTURE

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

Doctor of Philosophy in Engineering, Cyberspace Conc.

Supervisor of Dissertation Research

Head of Department

Department

Recommendation concurred in:

Advisory Committee

Approved:

Director of Graduate Studies

Approved:

Dean of the Graduate School

Dean of the College

ABSTRACT

Mobile devices can sense several types of signals over the air using different radio frequency technologies (e.g., Wi-Fi, Bluetooth, cellular signals, etc.). Furthermore, mobile devices receive broadcast messages from transmitting entities (e.g., network access points, cellular phone towers, etc.) and can measure the received signal strength from these entities. Broadcast messages carry the information needed in case a mobile device chooses to establish communication. We believe that these signals can be utilized in the context of access control, specifically because they could provide an indication of the location of a user's device. Such a "location proof" could then be used to provide access to location-based services. In this research, we propose a location-based access control (LBAC) system that utilizes tokens broadcasted by IEEE 802.11 (Wi-Fi) access points as a location proof for clients requesting access to a resource. This work differs from existing research in that it allows the verification of a client's location continuously and unobtrusively, utilizing existing IEEE 802.11 infrastructure (which makes it easily deployable), and resulting in a secure and convenient LBAC system. This work illustrates an important application of location-based services (LBS): security. LBAC systems manage access to resources by utilizing the location of clients. The proposed LBAC system attempts to take advantage of the current IEEE 802.11 infrastructure, making it directly applicable to an existing ubiquitous system infrastructure.

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

This dissertation is dedicated to my mother and father, Wejdan and Mahmoud Alamleh, who first taught me the value of education and critical thought.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Jean Gourd for the continuous support of my PhD study and research, for his patience, motivation, enthusiasm, and extensive knowledge. His guidance helped me in my of research and writing this dissertation. I could not have imagined having a better advisor and mentor for my PhD study.

Besides my advisor, I would like to thank the rest of my advising committee: Dr. Galen Turner, Dr. Jinyuan Chen, Dr. Manki Min, Dr. Mike O’Neal, and Dr. Pradeep Chowriappa for their encouragement, insightful comments, and guidance.

CHAPTER 1

INTRODUCTION

A smartphone is essentially a small computer that can fit in one’s hand. The beauty of using a smartphone lies in its ability to perform different functions and allow its users to access networks such as the Internet while providing convenience. In the past two decades, there has been rapid growth in the number of mobile devices utilized, especially with the global smartphone boom. Smartphones took over the wider consumer market, and as of 2018 according to the Pew Research Center, 77% of US adults use smartphones [50]. Smartphone users send, receive, and generate different types of data that can be consumed by functions and services. An example of such data is location data. This type of data is used in a location-based service (LBS).

A LBS is a service that takes the geographical location of a user’s device into consideration to provide certain application functionalities or services that are specific to a given location, such as ordering a ride or searching for nearby restaurants. LBSs have grown rapidly with the increase in the number of mobile devices and are expected to grow more among technologies such as the Internet of Things (IoT) and smart cities [46]. LBSs allow applications and services to be tailored to a given location, thereby limiting the services and/or functionalities a user receives to a physical location. LBSs tie data to a specific location where a user's activities happen and where services are offered. Location-related raw data can be useful and become a powerful source for data mining, where new information and patterns can be extracted to further improve the LBS or for different applications and services. For example, location and shopping pattern data can be used to advertise nearby businesses that may interest a user.

LBSs have various applications; for example:

1. Navigation: This service helps to locate the exact position of a user's device using a positioning system such as the Global Positioning System (GPS). It can also determine the best route for users to use to reach their destination. This is commonplace with smartphones. Apps such as Google Maps, Apple Maps, and Waze, for example, offer exciting features such as traffic conditions, reported accidents, and real-time updates to a route to assist users in reaching their destinations in the shortest time.
2. Emergency: One of the most vital applications of LBSs is for reporting emergencies. This is typically known as E911. Since LBSs reliably provide a user's location, it is easier for emergency services to track a user. Moreover, LBSs are very useful in disaster management, where they help to direct resources and to find the safest evacuation paths. A well-known example for using LBSs in emergency situations is the Amber Alert System, an emergency response system that broadcasts information about a missing person across U.S. interstates and highways in a location of relevance to the emergency.
3. Marketing and advertising: LBSs provide organizations the possibility of targeting customers based on their geographical location. This can also be used to study the behaviors of users and target advertisements that interest a specific group of users sharing some common interest.

4. Social media: LBSs can add a location element to user-generated content (e.g., images), providing context and potentially making it more interesting and relevant to social media users. For example, users now can “check-in” and geo-tag photos and videos that they post.
5. Workforce management: LBSs are used in workforce management by organizations that employ individuals out in the field or at multiple remote locations. LBSs allow employees to manage the business more efficiently by tracking the locations of their employees, allocating tasks based on their location, and keeping track of worker progress. A prominent example of this is used by shipping logistic companies to track customer package shipments to provide expected delivery dates and real-time package location information.
6. Gaming: Hobbies such as geo-caching (scavenger hunt-type games that use real geographical locations) have become popular over the last several years. A popular game, Pokémon Go, uses the location of players to provide a better gaming experience. Players can find Pokémon characters near them by navigating to a real physical location identified in the game.
7. Security: There are many security applications involving LBSs. An important one involves controlling access to resources for information security. This is known as location-based access control (LBAC). Secure LBSs can help to prevent unauthorized access to resources and to enhance a system's information security. In LBAC systems, an authorization entity typically verifies the location submitted by a user requesting access to a resource, and either grants or denies access to the resource based on the location. In general, LBSs make access to resources such as online banking more

secure [5]. An example of this is fraud prevention, where LBSs create an extra level of security by comparing a user's location to the location of a credit card transaction. Suspicious activities are then flagged or the transaction is declined.

There are many other applications of LBSs. Moreover, LBSs are becoming more common and widespread every day [44]. LBSs are also expected to advance and expand more with the recent developments in artificial intelligence, cloud computing, and IoT [60]. According to a Technavio report, the global indoor LBS market is going to witness a growth of more than 43% by 2020 [59]. As a result, LBSs have been (and continue to be) an interest for researchers and organizations.

In order to have an efficient LBS, the location information of a valid user is required. Location information can be obtained using different technologies. Most of today's smartphones are compatible with more than one location sensing method. Several of the most common methods used in obtaining a user's location in LBSs include:

1. GPS: The Global Positioning System is a radio navigation system that allows users to determine their exact location, velocity, and time anywhere in the world. GPS-based systems use signals from multiple satellites to calculate a user's position using the basic principles of trigonometry. Most smartphones contain a GPS “chip” (typically in the form of an integrated circuit). Despite GPS being a very efficient and convenient way to obtain a user's location, GPS-based systems have some weaknesses when used in LBSs. For example, GPS locations can be spoofed at a software level, thereby potentially rendering them unreliable and even insecure in LBAC systems. Furthermore, they suffer from performance limitations indoors, in that barriers such as buildings significantly reduce line-of-sight to satellites.

2. Cellular towers: A user's location can also be obtained through a network of cellular towers. A user's location can be determined by simply associating a cellular ID received at the user's device from the cellular tower broadcasting it, or by using triangulation and/or the angle of arrival (the angle in which the signal arrived the cellular tower's antenna). In general, the location obtained using cellular towers is more accurate (i.e., location errors are smaller) in urban settings due to the larger number of cellular towers located there. The location obtained through cellular towers is widely used in the E911 system as a backup to GPS [13]. This can occur if the caller's device does not support GPS or encounters GPS failure(s).
3. Wi-Fi and Bluetooth: Location obtained using Wi-Fi or Bluetooth is typically used for indoor applications. Wi-Fi networks usually consist of one or several wireless access points that allow devices to connect wirelessly to the network. A user's location obtained using Wi-Fi or Bluetooth is usually calculated in reference to a wireless access point transmitting a signal or by using indoor maps (which are visual representations of indoor geography). Determining a user's location indoors using Wi-Fi has been the target of several research projects as it is very important for emerging technologies such as IoT and robotics. Several techniques have been introduced that utilize different principles to calculate the location of devices [43], including the round-trip time (i.e., the time required for a wireless signal to travel back and forth between a user's device and a wireless access points), and the received signal power strength.
4. Sensors and wireless sensor networks: A wireless sensor network is a network of connected sensors that are used to monitor physical or environmental conditions such

as temperature, sound, and vibration. In some cases, wireless sensor networks can be used to calculate a user's location accurately. Different types of sensor networks can be used for this purpose: for example, ultrasonic sensor networks, light (i.e., photon-based) sensor networks, and others. However, most sensor networks are impractical to deploy due to their cost and complexity (they can consist of a large number of devices that need to be maintained and synchronized).

5. Hybrid positioning systems: In hybrid systems, different positioning technologies are combined. GPS is typically the main technology in these systems; however, other technologies are integrated to provide a way to overcome the limitations of GPS and to increase robustness of the overall system. For example, assisted GPS uses information from nearby access points to obtain a faster location fix. Hybrid positioning systems are increasingly being employed in LBS applications, particularly if they need to perform well in urban settings (e.g., to support self-driving cars).

Indeed, there are a number of ways to obtain a device's location. Consequently, location data can be presented in different formats such as:

1. Absolute location: A device's absolute location describes a fixed point on the earth. The most common way to identify the absolute location is by using a pair of coordinates (e.g., latitude and longitude). GPS typically provides an absolute location. This type of location data needs to be transmitted and stored securely to address privacy and security concerns.
2. Relative location: Refers to a position with respect to a reference point. Relative location can be expressed in terms of distance, travel time, or other forms. This type of location is usually utilized for indoor applications and can be used to generate

location proofs. A location proof is a digital certificate that indicates the position of a user at a specific time. Several researchers have studied generating relative location indoors using information from Wi-Fi or Bluetooth access points [30].

3. Location tags: Temporary keys generated and issued by an authorized entity at a specific location are known as location tags. Such tags are used to prove proximity to the entity issuing them and usually are valid for a limited time. Using location tags as location proof reduces the risk of revealing a user's location if the system is compromised since location tags do not carry the user's absolute location. Location tags can be installed at the location of interest in the form of QR codes, for example, or via near field communication (NFC) where users scan NFC “tags” to obtain the location tag. Moreover, they can also be broadcasted via Wi-Fi access points or Bluetooth.
4. Contextual data: This kind of data refers to the data that users can get from their surroundings. It can be used as a location proof; however, it must be verified by an authorized entity. Some examples of contextual data are the power footprints of Wi-Fi access points, lighting (i.e., photons), etc. For example, Wi-Fi channel characteristics are used for paring two devices [55], and light sensors are used in location-based illumination control [29].

Location can either be calculated at a user's device or by other nodes in a system that can calculate a user's location or verify a user's proximity. As discussed above, there are several methods that can be used to obtain a user's location, and these methods vary in accuracy, efficiency, and cost. Moreover, they provide different types of location data (resulting in location proofs). Thus, the location method to be used in a LBS depends on the LBS and what method fits it the best based on its unique requirements.

One important use of LBSs is LBAC. In this dissertation, we propose a LBAC system that utilizes ubiquitous Wi-Fi access points already in a location to manage access to resources. A significant aspect of this work is that the proposed system is unobtrusive (i.e., there is zero interaction on the part of users in the system).

LBAC is an information security methodology that utilizes the location of users to control who can access resources. LBAC performs authentication to prove the identity of a user and subsequently grant (or deny) permission to access specific resources by verifying the user's presence at a distinct location. In order to produce an effective LBAC system, the user must present a location proof that verifies the user's physical presence at a distinct location. Moreover, an authorization entity must be able to verify the location accurately. Based on that result, it grants or denies access to resources. An extension to this involves managing which specific resources a user can access based on the user's location. It is also important for a LBAC system to be able to limit access to resources during the time that a user is physically present at an appropriate location (i.e., terminating access once the user leaves).

LBAC allows for the implementation of location-based security policies. Examples of this include limiting access to sensitive data when a user is located at a specific location. Furthermore, LBAC reduces the malicious use of information because access is limited to specific location(s). For example, bankers have no need for confidential customer data outside of their offices; therefore, access to this data from the outside is most likely malicious and should, therefore, be prevented.

With the increase of LBSs, privacy issues have been raised. Moreover, they have become significant with recent breaches to major technology companies such as Facebook, Yahoo, and Equifax [20]. A user's location (and a location proof) is sensitive information that must be protected from unauthorized access so that it is not misused. Privacy is a very important goal when designing a LBS. Generally, there are two types of data that are relevant to LBSs that must be kept private: the actual location data and the user's personal data. Studying the subject of privacy in LBSs has become important to many researchers in order to find ways to maintain privacy and prevent unauthorized access to data in LBSs [3]. As a result, various anonymization and obfuscation algorithms have been developed to achieve this goal. An example is the use of obscure representations of location data to improve privacy [37].

In order to have an effective LBAC system, it must be convenient for users of the system. This typically includes ensuring that it does not require extra effort from users in the system. User convenience is a key factor in the success of any system [39]. It is known that complex systems that require extra user effort tend to drive them away [39]. However, from a deployment point of view, systems must be affordable, easy to deploy, and should not require significant infrastructure (in fact, it is typically best if a system makes use of existing infrastructure, especially if it is to be widely used). Building a secure and inexpensive LBAC system is essential.

The importance of LBSs in the present day is clear, with many applications in different areas ranging from business to public safety, and security. There are different methods that can be utilized to obtain a user's location, and each has strengths and weaknesses. The need to have LBSs that do not compromise user privacy and security, and have the ability to verify location claims efficiently and accurately has emerged. Building an efficient and secure LBAC system that can be implemented for use in different LBSs is the goal of this research. Specifically, this work aims to explore the possibility of building a secure, inexpensive, unobtrusive, and robust LBAC system utilizing existing ubiquitous infrastructure.

Today, a username and a password are typically used for authentication. Our goal is to add a second authentication factor for more security and to add LBS compatibility, in a way that it is secure, unobtrusive for the users, inexpensive and utilizes existing infrastructure. To accomplish this, we propose to utilize the current IEEE 802.11 infrastructure and potentially making slight modifications to existing access points. Furthermore, a goal of the proposed system is to continually verify the presence of a user in the context of authenticating access to resources. We call the proposed system a zero-effort system, which means that a user's location proofs are collected and sent automatically without adding any extra burden on the user.

1.1 Technical Background

A LBAC system consists of two components: a location component that provides a user's location for authentication, and an access control component that controls access to resources based on the location component provided by the user. In this section, necessary background will be covered before we introduce our proposed system in a later chapter.

Access control in information security is the process by which users are granted access and certain privileges to systems, resources, or information. In general, access control includes identification, authentication, and authorization. Identification refers to the process of a user claiming an identity (i.e., “Who are you?”). This is done, for example, by providing a username. Authentication is the process of proving the identity of a user by verifying credentials that were provided in the identification process (i.e., “Are you who you say you are?”). Authorization is ultimately making the decision to grant or deny access based on the outcome of authentication.

The process of authentication can be performed as single-factor or multi-factor. In single-factor authentication, one credential is used to confirm a user's claimed identity. In multi-factor authentication, multiple credentials are used: for example, a user providing a password and a second factor such as a fingerprint in two-factor authentication. Second factors include things like security tokens or biometric factors such as a fingerprint, voiceprint, or retina scan. Multi-factor authentication adds an extra layer of security and prevents attackers from gaining unauthorized access in cases where the first factor is compromised.

A user's location can also be used as a factor (including as a second factor) for authentication in access control applications. In this way, access is granted only if a user is in a specific physical location. This can be done by having the user provide location information manually, through a mechanism on a device (e.g., via an installed application on a smartphone), or via a location proof that can be in the form of a key that indicates that a user is close to a node requesting authentication to a resource. The authorization entity verifies this information and makes a decision to grant or deny access based on it. In this work, we explore the possibility of modifying existing artifacts in IEEE 802.11 access points to accurately provide location proofs for users in a LBAC system.

IEEE 802.11 refers to the set of standards that defines communication for wireless local area networks (WLAN). Generally, IEEE 802.11 is known as Wi-Fi. The IEEE 802.11 protocol allows computers to communicate wirelessly with each other. There are different versions of the IEEE 802.11 standard: for example, IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n, and, recently, IEEE 802.11ac and IEEE 801.11ad. These different specifications primarily represent changes in speed, performance, and range. As new Wi-Fi technology emerges, standards are improved and released to the public.

WLANs operate on one of two major frequencies: the 2.4 GHz band or the 5 GHz band. A band is a specific range of frequencies that is regulated by a government for a specific use. In IEEE 802.11, bands are divided into channels that allow various nearby devices to share the same frequency without resulting in (or limiting) interference. The 2.4 GHz band has 11 channels and is more subject to interference, as it is a general use band that is utilized by many other applications such as Bluetooth, microwave ovens, baby monitors, cordless (hard line) phones, and so on. In this band, there are only a few channels that can be used at the same time without interference. The 5 GHz band has 19 usable channels. Devices using the 5 GHz band must be able to examine a channel's usage and choose one that minimizes interference. In general, the 5 GHz band provides faster data transfer rates at a shorter distance, whereas the 2.4 GHz band offers coverage at farther distances but may perform at slower speeds. Most of the access points today run in dual mode, providing both 2.4 GHz and 5 GHz coverage, simultaneously.

Data transmission speeds vary based on the version of the Wi-Fi protocol that is used. It ranges from 6 Mbps (megabits per second) in IEEE 802.11a to 600 Mbps in IEEE

802.11n. The coverage range of WLAN access points varies from a few meters to 100 meters. An access point's range depends on the protocol used, its transmitter type (e.g., antenna type) and its configurations (e.g., antenna transmission power settings).

In this work, we propose to use existing the IEEE 802.11 access points to broadcast meaningful tokens that can be used to generate a location proof. In general, a token is a secret key that is used to gain access to a digital resource. Our first step is to determine if it is possible to broadcast tokens in the context of Wi-Fi protocol constraints. Although there are several ways that we could go about broadcasting tokens, we propose to manipulate the beacon frame of network packets in WLANs. Fundamentally, a frame is a base unit of digital transmission in networks. The beacon frame is one of the management frames in the IEEE 802.11 protocol. It is used to announce the existence of a network, and it plays an important role in many network maintenance tasks. Beacon frames are transmitted at regular intervals (called the beacon interval), allowing network interfaces on user devices to find and identify networks. They also carry any necessary parameters for joining the network. Access points are responsible for transmitting beacon frames. The area in which beacon frames appear defines the basic service area, which is the area where an access point can provide the service of providing wireless access to the network. Devices need the information in the beacon frame to be able to connect to an access point; therefore, mobile devices must be close enough to access points to be able to receive the beacons. In general, the beacon interval of most access points is set to 100 milliseconds by default. However, it can be adjusted in most access points. Access points must schedule beacon transmission at the set beacon interval, but the transmission may suffer some delays to avoid collision with beacon frames broadcasted by other devices and access points in the area.

As shown in Figure 1-1, the structure of a beacon frame in IEEE 802.11consists of the following:

1. MAC (Medium Access Control) header: The MAC header includes source and destination MAC addresses. A MAC address is a hardware identification number that uniquely identifies each device on a network. The destination address in broadcasted messages is always set to the MAC address (FF:FF:FF:FF:FF:FF), which is the default broadcast MAC address in computer networks. This forces all other devices on the applicable channel of the frequency band to receive and process each beacon frame.

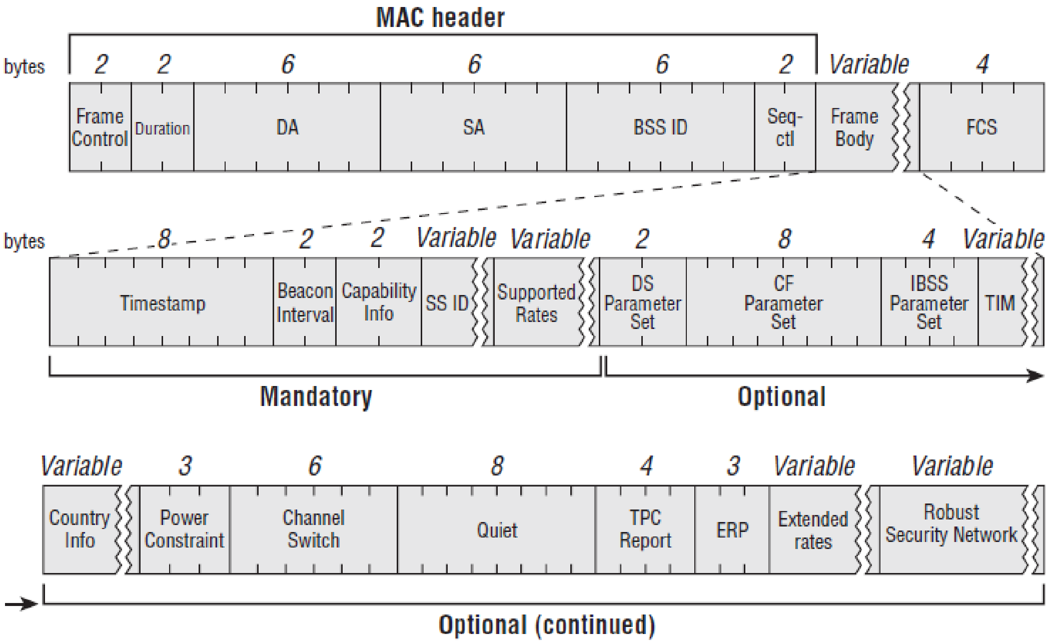


Figure 1-1: The structure of the IEEE 802.11 beacon frame [25].

2. The frame body, which includes the following:

a. A timestamp field that is used by a network device to update its local clock. This process enables synchronization among all devices that are associated with the same access point.

b. A beacon interval field that represents the amount of time between beacon transmissions, and it is used to determine the next time to check for incoming traffic in case the device goes into power saving mode, which allows a device to switch its Wi-Fi radio unit on and off several times per second to extend the battery's life.

c. A capability information field that advertises the network's capabilities and requirements to devices that wish to connect to the WLAN.

d. A Service Set Identifier (SSID) that uniquely identifies a specific WLAN. The SSID is simply the name of a Wi-Fi network. Before associating with a particular WLAN, a device (known as a Wi-Fi NIC – a wireless network interface card) must pair with the SSID of the access point. By default, access points include the SSID in the beacon frame to enable functions on devices to identify the SSID and automatically configure the NIC with the proper SSID. However, most access points have the ability to omit the SSID from beacon frames, thereby hiding the network from devices.

e. Supported rates describe the speeds (in Mbps) that a particular WLAN supports, which help devices to stay within speed limits and avoid using unsupported rates. Devices can use the supported rate information to decide which access point is more suitable for connection.

f. A parameter set field that includes information about the specific signaling methods to be used when associating with an access point (e.g., the spread spectrum modulation technique, which is issued to reduce overall signal interference). The beacon frame includes the appropriate signaling methods to adapt to the channel's condition and achieve efficient data transmission.

g. A traffic indication map (TIM) that provides a list of all devices that have undelivered data buffered on the access point (i.e., that is waiting to be delivered).

3. A frame check sequence (FCS) provides error detection capabilities.

Since this work aims to utilize the IEEE 802.11 beacon frame to transmit tokens for the authentication of clients to resources, different fields in the beacon frame could be modified to carry the token. In fact, several fields may be unnecessary during normal use and can be exploited for use in the proposed LBAC system. Access points currently available in the market typically transmit power in the range of 16 to 32 decibel milliwatts (dBm). A dBm is the power ratio referenced to 1 milliwatt (mW) in the logarithmic scale. This scale is used as a measure of power in communication systems because of it provides a convenient way of referring to ratios, such as the ratio between the amplitudes of a transmitted signal and a received signal. As a wireless signal travels, it decays before arriving at a receiving device. This can be expressed using the Friis Transmission equation:

Eq. 1-1

$$\frac{P_r}{P_t} = D_t D_r \left(\frac{\lambda}{4\pi d}\right)^2$$

Where

P_r is the power at the receiving antenna.

P_t is the power transmitted by the transmitting antenna.

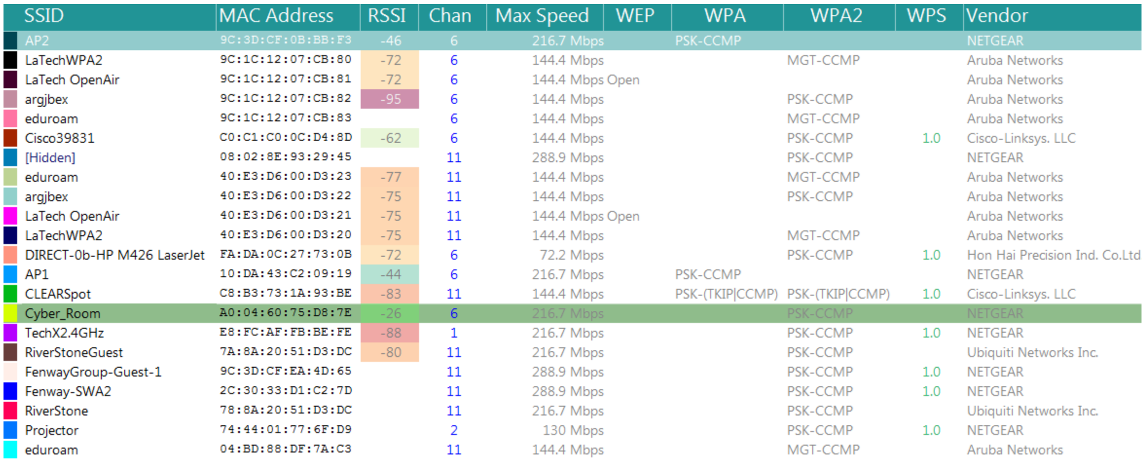
D_t is the directivity of the transmitting antenna. Antenna directivity is a parameter that measures the degree to which the radiation emitted by the antenna is concentrated in a single direction.

D_r is the directivity of the receiving antenna.

λ is the wave length in meters. For example, the wave length for 2.4 GHz Wi-Fi is approximately 0.125 meters and for 5 GHz Wi-Fi is approximately 0.0625 meters.

d is the distance between the transmitting and receiving antenna in meters.

As can be seen from Equation 1-1, the power of the received signal at the receiving antenna is inversely proportional to the square of the distance that the signal traveled. Thus, the received signal strength at a device is consistent with its location in reference to the access point. Figure 1-2 shows the beacon information received by a mobile device on the campus of Louisiana Tech University. The figure shows Wi-Fi access point information obtained in real-time by an application known as Acrylic Wi-Fi Home [1]. Information provided includes the access points in range of the device and their corresponding Received Signal Strength Indicators (RSSI) in dBm. The application collects information about access points scanned by a network device. As can be seen in the figure, the RSSIs have negative values. A negative dBm means that a negative exponent is applied in power calculations (e.g., 0.1 mW equals -10 dBm is, similarly 0.01 mW equals -20 dBm, etc.).



SSID	MAC Address	RSSI	Chan	Max Speed	WEP	WPA	WPA2	WPS	Vendor
AP2	9C:3D:CF:0B:BB:F3	-46	6	216.7 Mbps		PSK-CCMP			NETGEAR
LaTechWPA2	9C:1C:12:07:CB:80	-72	6	144.4 Mbps			MGT-CCMP		Aruba Networks
LaTech OpenAir	9C:1C:12:07:CB:81	-72	6	144.4 Mbps	Open				Aruba Networks
argjbex	9C:1C:12:07:CB:82	-95	6	144.4 Mbps			PSK-CCMP		Aruba Networks
eduroam	9C:1C:12:07:CB:83		6	144.4 Mbps			MGT-CCMP		Aruba Networks
Cisco39831	C0:C1:C0:0C:D4:8D	-62	6	144.4 Mbps			PSK-CCMP	1.0	Cisco-Linksys. LLC
[Hidden]	08:02:8E:93:29:45		11	288.9 Mbps			PSK-CCMP		NETGEAR
eduroam	40:E3:D6:00:D3:23	-77	11	144.4 Mbps			MGT-CCMP		Aruba Networks
argjbex	40:E3:D6:00:D3:22	-75	11	144.4 Mbps			PSK-CCMP		Aruba Networks
LaTech OpenAir	40:E3:D6:00:D3:21	-75	11	144.4 Mbps	Open				Aruba Networks
LaTechWPA2	40:E3:D6:00:D3:20	-75	11	144.4 Mbps			MGT-CCMP		Aruba Networks
DIRECT-0b-HP M426 LaserJet	FA:DA:0C:27:73:0B	-72	6	72.2 Mbps			PSK-CCMP	1.0	Hon Hai Precision Ind. Co.Ltd
AP1	10:DA:43:C2:09:19	-44	6	216.7 Mbps		PSK-CCMP			NETGEAR
CLEARSpot	C8:B3:73:1A:93:BE	-83	11	144.4 Mbps		PSK-(TKIP|CCMP)	PSK-(TKIP|CCMP)	1.0	Cisco-Linksys. LLC
Cyber_Room	A0:04:60:75:D8:7E	-26	6	216.7 Mbps			PSK-CCMP		NETGEAR
TechX2.4GHz	E8:FC:AF:FB:BE:FE	-88	1	216.7 Mbps			PSK-CCMP	1.0	NETGEAR
RiverStoneGuest	7A:8A:20:51:D3:DC	-80	11	216.7 Mbps			PSK-CCMP		Ubiquiti Networks Inc.
FenwayGroup-Guest-1	9C:3D:CF:EA:4D:65		11	288.9 Mbps			PSK-CCMP	1.0	NETGEAR
Fenway-SWA2	2C:30:33:D1:C2:7D		11	288.9 Mbps			PSK-CCMP	1.0	NETGEAR
RiverStone	78:8A:20:51:D3:DC		11	216.7 Mbps			PSK-CCMP		Ubiquiti Networks Inc.
Projector	74:44:01:77:6F:D9		2	130 Mbps			PSK-CCMP	1.0	NETGEAR
eduroam	04:BD:88:DF:7A:C3		11	144.4 Mbps			MGT-CCMP		Aruba Networks

Figure 1-2: Beacon information from several access points on campus [1].

Most of the access points sold on the market contain built-in firmware that allows some degree of configuration. Such firmware is typically Linux-based. Linux is a free, open-source operating system. It is modifiable by anyone, and variations of the source code (known as distributions) can be freely created. The most common use of the Linux operating system is as a server (a machine dedicated to providing services to people: for example, a Web server that provides Web pages). However, Linux is also used in desktop computers, smartphones, network routers, gaming consoles, etc.

The Linux operating system supports a command line user interface called the shell. The shell allows controlling the system via a text-based command line and supports writing programs that can be used by the operating system to control parts of the system. Bash is an improvement of the shell that extends it. One of the most important tools available in the Linux shell is the Secure Shell (SSH). SSH is a network protocol that allows strong authentication and encrypted data communications between two computers connecting over an open network such as the Internet. SSH allows remote logins and supports remote command execution.

While it is possible to use SSH with a username and a password, SSH relies more securely on public-private key pairs to authenticate hosts to each other. A key in cryptography is a variable value that can be used as an authentication factor or to encrypt or decrypt a message. Moreover, a key is used in hash-based message authentication code (HMAC) functions. A hash is a function that maps data of arbitrary size to data of a fixed size. Some well-known examples of hash functions include MD5, SHA1, and SHA256. They are essentially math functions that produce a (hopefully) unique hash for arbitrary-sized data. HMAC utilizes a key along with a hashing function to generate a message authentication code that is used for several applications such as verifying the data's integrity and generating time-based passwords. In this research, HMAC is used to generate time-based tokens, which is discussed in detail in Chapter 4.

1.2 Conclusion

In this chapter, LBSs, their applications, and different methods to obtain the location of mobile devices were discussed. Furthermore, LBAC systems were introduced, along with their applications. In the next chapter, we will discuss existing research in the field of LBAC.

CHAPTER 2

BACKGROUND

In this chapter, we survey prior attempts to leverage user location for access control purposes. Existing research includes LBAC using GPS, wireless sensor networks, Bluetooth, IEEE 802.11, and others.

2.1 Related Work

This section provides an analytical overview of the significant literature published in the LBAC topic, which is categorized based on the location method used.

2.1.1 LBAC Systems That Use GPS for Localization

GPS is the most straightforward way to obtain a user's location on GPS-enabled devices. Most smartphones are equipped with GPS. In 2006, Takamizawa and Kaijir [in 57] proposed an authentication method for distance learning applications by using cellular phones as an authentication token by using functions available on the phone (e.g., camera, GPS, etc.) for authentication. Furthermore, they proposed to potentially use a phone's GPS functionality (if it was available). Three years later the same authors [in 56] proposed using the location of a cellular tower as a second factor for authentication. The weakness in these related works is that, in most devices, the GPS location can be spoofed – either in hardware, through the operating system, or even at the application level [28]. From our perspective, this is a significant weakness.

In 2012, Zhang, Kondoro, and Muftic proposed a hybrid approach for location-based authentication and authorization using smartphones [69]. Their approach proposed using two sets of location inputs obtained from two different location sources: the IP address of the client and the MAC address of a nearby access point with the strongest signal. The weakness in this method is that it requires two sets of location inputs from two different types of sources, which may be difficult to obtain on all devices. Moreover, the method also suffers from the potential for spoofing GPS, using a network proxy to spoof the IP address, or the introduction of a rogue access point with a spoofed MAC address. This renders the method vulnerable and inconsistent.

Utilizing GPS as a proof of location inherently has weaknesses. Not all devices that users utilize (particularly mobile devices) have GPS built into them (e.g., laptops and tablets typically have no such support); therefore, LBAC systems that use GPS location for location proofs and/or access control fail to provide services to users of these devices. Another weakness inherent with GPS is that it suffers from performance limitations when devices are indoors. There are often too many physical barriers to get precise location fixes using satellites. It is therefore quite challenging to achieve a precise location proof indoors using GPS.

2.1.2 LBAC Systems That Use Wireless Sensor Networks for Localization

There is a wide variety of sensor types that can be used in wireless sensor networks. This has led to several techniques for user localization using devices in wireless sensor networks. In [45], techniques of localization in wireless sensor networks are surveyed: angle of arrival measurements, distance-based measurements, and RSS based measurements. There have also been hybrid systems that use both wireless sensor networks and WLANs for localization. The authors [in 21] propose using both time and power measurements of devices in reference to three sensor nodes. The authors [in 12] proposes RSSI-based indoor localization for wireless sensor networks in smart homes. Furthermore, there have also been hybrid systems that use both wireless sensor networks and WLANs for LBAC. For example [51], which specifies authorization based on both role and location in a WLAN with assistance from a sensor network. However, using wireless sensor networks for LBAC requires additional infrastructure (the embedded wireless devices), and often, these devices must be more capable than simple routers or switches in WLANs. Therefore, this method can be costly to deploy and maintain.

2.1.3 LBAC Systems That Use Bluetooth for Localization

Most smartphones today are Bluetooth capable. Bluetooth is a standard for short-range wireless communication that allows portable devices such as cell phones and laptops to communicate with each other. Its range is typically less than 10 meters. Inherently, this is a weakness in that the distance between devices that utilize Bluetooth for communication must be small. There is existing research that has studied utilizing Bluetooth for localization. For example, the authors [in 41] propose using Bluetooth Low Energy (BLE) for indoor localization. Vascak and Savko [in 62] propose using BLE for indoor localization and navigation. Furthermore, there is existing research that has studied utilizing Bluetooth for LBAC applications. One such work proposed to use Bluetooth devices to broadcast location proofs. Jansen and Korolev proposed a location-based authentication mechanism that employs policy beacons [33], which are small Bluetooth devices placed in an area where a distinct policy is in effect. This was done so that a user's device could establish proximity to the beacons.

Similarly, Grumaz [in 26] proposed to use multiple Bluetooth token devices to transmit keys to be used by Bluetooth-enabled devices for access management purposes. Also, Van Rijswijk-Deij [in 61] proposed using Bluetooth to transmit a one-time password. In this work (as shown in Figure 2-1), a mobile application installed on a smartphone scans for the one-time password beacon that is transmitted by a Bluetooth-enabled device. It then uses that password with a key saved on the device to authenticate to a server (providing some service or resource). The weakness in these approaches is that Bluetooth typically has a short range. It is, therefore, challenging to deploy the system for LBAC systems in large areas or for devices that are not always close to users wishing access to resources. Moreover, Bluetooth requires additional hardware that is not always guaranteed to be in every user's device.

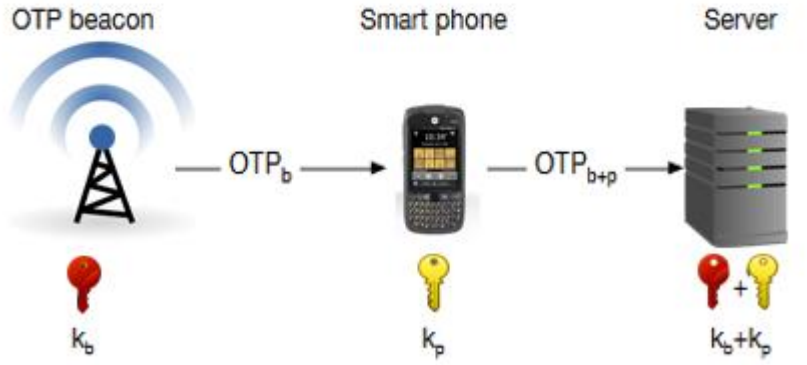


Figure 2-1: Bluetooth Onetime password system's overview [61].

Other Bluetooth-based LBAC systems were introduced ([14][23][58]) that employ knowledge of neighbors. In these systems, a user (called a claimer) authenticates by proving proximity to trusted users (called verifiers). The verifiers are in charge of creating location proofs that the claimer uses to prove its heir location to the authentication server. However, the problem with this method is that it requires the participation of verifiers that need to register to the authentication server and be at a specific location. Moreover, these LBAC systems can be compromised if a verifier acts maliciously.

2.1.4 LBAC Systems That Use IEEE 802.11 Access Points for Localization

Because IEEE 802.11 systems and devices are ubiquitous (both indoors and, to some degree, outdoors) and are becoming a primary way to access large networks such as the Internet, there has been a lot of research that has studied utilizing IEEE 802.11 as a means of localization, particularly indoors (see [6,31,43,64]). Some of these have also been applied in LBAC applications. Specifically, this is done using channel characteristics to derive a location proof, or by having the IEEE 802.11 access points provide the location proof. Since this represents a large body of existing work, each of these will be discussed below.

2.1.4.1 *Channel characteristic-based IEEE 802.11 systems*

Using this method, location proofs are derived by using signal information received from access points. Such information includes SSID, MAC address, and RSSI – all being broadcasted by surrounding access points. These systems are used for applications such as pairing co-located devices [2] or to protect against wormhole attacks in services like Samsung pay [52]. In some systems, location proofs are derived from the physical layer information for signals received from the surrounding access points. This information includes multipath profiles [as in 65], which relies on the similarity of multipath profiles for co-located users. Moreover, location proofs are derived using mathematical modeling on channel characteristics; for example, [in 24] a statistical model is used on RSSIs, [in 27] neural network decisions are used after training the model using RSSIs, and [in 34] a fuzzy extractor is used on channel characteristics and RSSIs. Moreover, the authors (in [67]) proposed generating a public location tag using RSSIs, packet sequence numbers, and MAC addresses of the ambient access points for proximity detection, the authors (in [70]) proposed using a bloom filter and a fuzzy extractor on IEEE 802.11 frames and the control messages in 4G LTE network to derive location proofs. A well-known example is Amigo [in 55], a technique that authenticates co-located devices using knowledge of their shared radio environment as proof of physical proximity. Co-located devices tend to have similar characteristics. In the proposed technique, the mobile device collects information (including RSSI values) from nearby access points. A classifier is then trained with data that indicates that two devices are co-located if they are five centimeters apart. Then, the proof of physical proximity is used to securely pair co-located devices.

The authors [in 36] proposed generating a location proof by using the wireless signal information of a user's current location to authenticate to a server which then verifies the location proofs using a database of available Wi-Fi stations at different locations. However, this method requires frequent updates to the database in order for the system to operate properly. In general, it is challenging to use methods that utilize channel characteristics in client-server-based applications, because in these systems, the server must be able to verify the location proofs. Consequently, it must have real-time access to the channel characteristic used in the generation of the location proofs, which is challenging for systems that use a remote verification server.

2.1.4.2 *Token-based systems*

In token-based systems, the access points are in charge of producing a token that can be used as a location proof. In 2006, Cho, Bao, and Goodrich proposed a system to control access to a WLAN infrastructure based on IEEE 802.11 devices [15]. As shown in Figure 2-3, areas, where network access was to be granted (called access groups), were defined by the overlapping coverage of the access points. In the proposed system, the mobile device collects nonces from multiple access points that are collectively used to define an access group. A nonce is an arbitrary number that can be used just once in cryptographic communication. The mobile device then derives a location key based on the nonces that is used to authenticate and connect to one of the access points. The weakness in this system is that it assumes the access points to be interconnected through a secure channel in order to exchange their nonces (without specifying how) so that any access point in their proposed system could authenticate users. This usually entails extra connections and hardware that complicates the system. Moreover, a user device may only connect to the network through access points that have access to the nonces. On the other hand, in their proposed system, access areas (i.e., the area where access to resources is to be granted) are specified by whether the user's device receives the access point's broadcast. Usually, devices can receive broadcasts from extended ranges. This results in difficulties in defining smaller access areas.

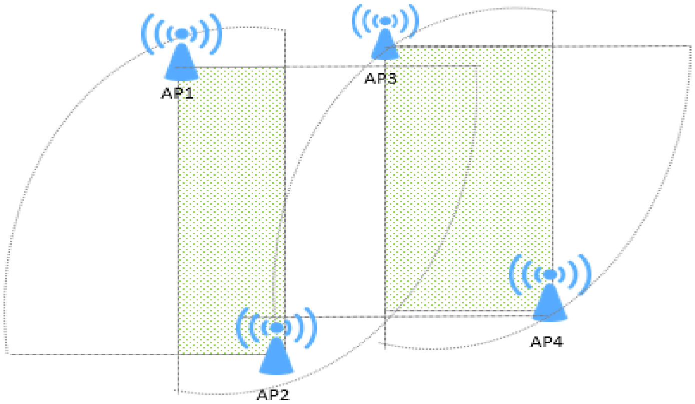


Figure 2-2: Defining areas granted network access [15].

A similar system is proposed by the same authors [in 8] and involves the use of a key server as shown in Figure 2-4. The key server manages the distribution of keys to the access points. The mobile device collects the keys from the access points and submits an access request through the network infrastructure. The key server verifies the request and makes the authorization decision to grant or to deny access. The user must use the same network infrastructure in order to connect to the key server for authorization. The weakness in this system is that a key server is needed, thus adding extra hardware and cost. On the other hand, similar to the system above, access areas are specified by whether the user's device receives the access point's broadcast, which results in difficulties in defining smaller access areas.

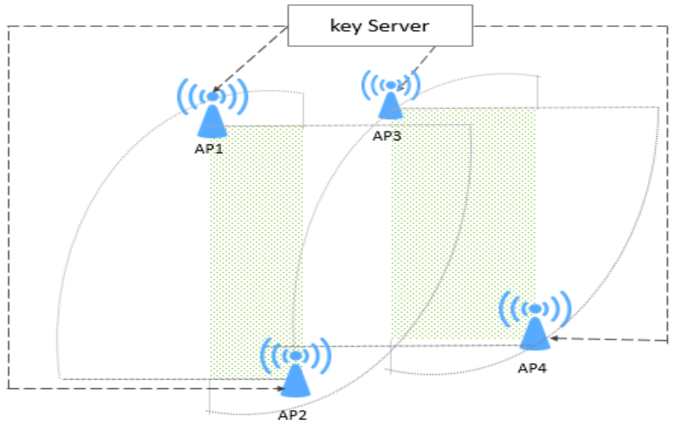


Figure 2-3: The key server distributes keys to the access points [8].

Huseynov and Seigneur [in 32] proposed using the SSID field in the beacon frame to provide a one-time password (OTP). The OTP is used as a second authentication factor when authenticating to a remote server. The major weakness here is that it does not truly utilize localization. It is merely a proximity-based system; therefore, any users who are connected to the access points (and the key server) may be granted access. Moreover, the proposed system does not continue verifying a user's presence once access is granted. Thus, if a user should leave the area, access would not necessarily be terminated. In order to have secure LBAC, continuous verification of a user's location is necessary.

Bailey and Brainar proposed sending authentication data using the SSID field by setting two unidirectional channels as shown in Figure 2-6 [7]. Their One-touch Financial Transaction Authentication system uses a token device and the user's device to communicate and send the authentication data via broadcasting SSIDs. However, their proposed system is bi-directional and it occupies the user's Wi-Fi interface to send the authentication data by broadcasting SSIDs. Thus, the Wi-Fi interface on the user's device is occupied and cannot be used for network connection functionalities.

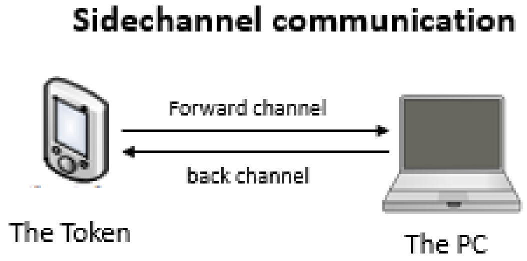


Figure 2-4: One-touch Financial Transaction Authentication system overview [7].

There is also other research that involves utilizing wireless access points to issue location proofs for devices. In [35], a user requests a location proof from the access point, which provides it and then sends it to a server that is connected to a verifier. In their proposed system, the verifier receives location claims from users and subsequently verifies them. In [42], a third-party server and an access point work together to issue a location proof for a user requesting it. Then, the user uses this location proof to get access to a LBS. In [53], access points send out beacons advertising their support for location proofs. A user then requests a location proof by sending information about the beacon frame received. Subsequently, the access point verifies the user's request and provides the user with a location proof.

In general, the location proofs issued to the user's device certify the user’s presence near the access point. Moreover, a location proof issued by an access point can be used by the user's device to prove the current location or a past location. In general, in these systems, the device requests a location proof from an access point. The access point then verifies the request and issues a location proof. However, the users must be connected to the access point in order to receive location proofs. As a result, clients lose their freedom to choose the connection to access the desired network on which the requested resource(s) reside. Another weakness is that this method puts the responsibility of providing connection to the desired network on the access points providing location proofs, which also put the burden on the user to be connected to the access point that sends location proofs in order to be able to get one.

Pandey, Anjum, Kim, and Agrawal propose a scheme based on the current access point's capability of transmitting at different power levels [48]. In this work, multiple access points transmit nonces at different power levels. As shown in Figure 2-7, each region has its unique set of nonces based on how far the region is from the access points. A user's device responds by retransmitting the set of messages it received to the access point it is associated with. These messages are then used to determine the location of the user's device. This proposed system's main purpose is localization; it does not have an access control element; moreover, it does not study the possibility of using the nonces for LBAC applications. In addition, the system requires a separate controller (an extra computing unit) that is connected to all of the access points.

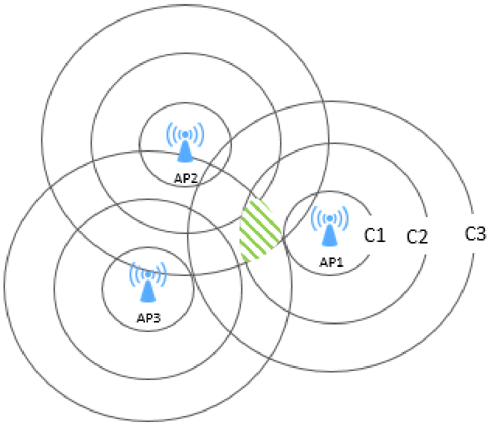


Figure 2-5: Three access points transmitting nonces at three different power levels [48].

2.1.5 LBAC Systems Using Other Technologies

This subsection discusses related research that proposes other technologies for LBAC uses.

2.1.5.1 *Near Field Communication (NFC)-based systems*

NFC is a short-range wireless connectivity standard that uses the magnetic field induction to enable communication between devices that are very close (i.e., within a few centimeters of each other). LBAC systems that utilize NFC typically require the user to be very close to NFC devices because they operate at short range. Xin-fang, Ming-wei, and Jun-ju [in 68] present a LBAC system to protect data security in mobile storage devices by embedding an RFID tag that makes it only possible to read the encrypted files on the mobile device when a device is at specific locations. Berbecaru [10] proposed using a token received by an NFC device as an authentication factor. Avdyushkin and Rahman [in 4] proposed using NFC along with a Wi-Fi access point to validate the location of users. In their proposed system, a unique identifier is obtained from an NFC tag installed in an area and scanned by the user's device. Subsequently, the user's device sends it to the server using a proximate access point. The server then verifies the user's location by checking if the tag was received from a proximate access point. However, LBAC systems using NFC require additional hardware and NFC capable mobile devices. Even today, many mobile devices do not support NFC (i.e., this is not currently guaranteed). Therefore, coverage may be limited, and users not possessing NFC-enabled devices would not be able to use the system.

2.1.5.2 *Cellular tower-based systems*

Wullems, Looi, and Clark proposed location-based auditing and access control by using the global cellular ID, which the user's device is connected to, and by using the timing advance measurements (i.e., the time taken for the signal to travel between the mobile device and cellular tower) to calculate the mobile device's location [66]. The weakness in this approach is that cellular towers do not provide a location that is accurate enough for precise localization; therefore, it is not always suitable for access control applications.

2.2 Conclusion

As seen from the related work, there is relatively little that addresses LBAC in an inexpensive and deployment-ready way, and that is unobtrusive for users. Most of the proposed work includes extra authentication steps, which is inconvenient for the users, or includes additional hardware and networking requirements. Moreover, most of the research addresses either location fixing (i.e., where the user is) or access control (i.e., is the user authorized to access a resource). However, we believe that it is important to address both as a whole by providing a unified method to address location fixing and access control simultaneously, and in an unobtrusive way using the current ubiquitous IEEE 802.11 infrastructure. This has the potential to result in an efficient, inexpensive, and secure LBAC system.

CHAPTER 3
THE PROPOSED SYSTEM

In this chapter, we introduce our proposed LBAC system architecture. The overall architecture is discussed in Section 3.1, and the design requirements for an efficient LBAC system are discussed in Section 3.2. The design of our proposed system is introduced in Section 3.3, and the challenges in achieving this design are discussed in Section 3.4. We conclude in Section 3.5.

3.1 LBAC Architecture

Conventional access control systems rely on the identities of users to determine access rights. In LBAC systems, however, access rights are additionally determined by the locations of the users. An example of such a need is limiting access to sensitive information when employees are located in a building or a room at a workplace. In LBAC, not only is a user's identity verified, but the user's location is also required. Once verified, the user is granted access to a particular resource that corresponds to the access policy in place. The access policy outlines the users of the system and their corresponding access rights to different resources provided by the system. Thus, in LBAC systems, the identities of the users along with their physical location play a role in determining what resources they are granted access to. Typical LBAC authentication procedures are shown in Figure 3-1. In the system shown in the figure, it is assumed that able to verify it. This is unlike other LBAC systems, where some other node in the network (e.g., an access point, controller, etc.) is in charge of generating and/or verifying location proofs. The authorization entity is in charge of verifying the user's credentials and authorizing access to requested resources. The steps for authentication and authorization in a LBAC system are as follows:

1. A user sends his/her identity and requests access to resources.
2. The authorization entity requests the credentials that prove the user's identity. These credentials are dependent on the number of authentication factors. For single-factor authentication, the authorization entity only requests a location proof. However, for multi-factor authentication, the authorization entity additionally requests other credentials (e.g., a password, biometric identifier, etc.) along with the location proof.

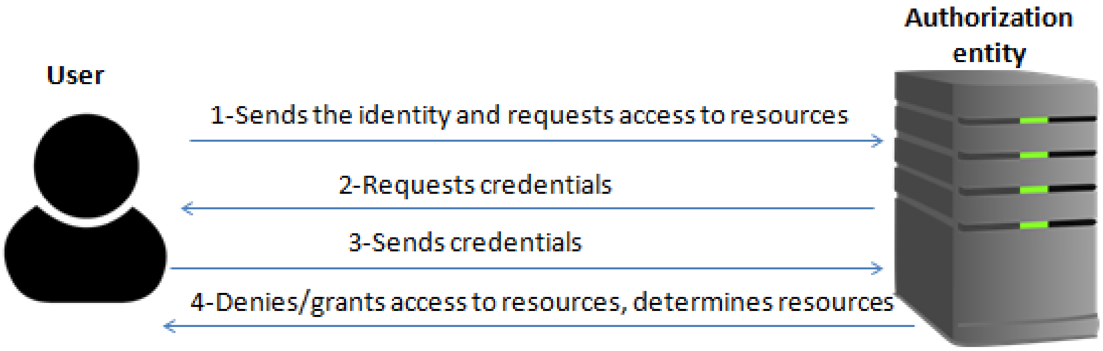

User
Authorization entity
1-Sends the identity and requests access to resources
2-Requests credentials
3-Sends credentials
4-Denies/grants access to resources, determines resources

Figure 3-1: Typical LBAC authentication process.

3. The user responds with the credentials that include a location proof to the authorization entity.
4. The authorization entity verifies the received credentials, including the location proof. Based on that information, it grants or denies the user access to the requested resource(s). If access is granted, it determines the resources that the user has access to

based on the user's identity and location. To accomplish this, the authorization entity maintains an access policy that contains all user identities in the system and their access rights at each distinct physical location. Access to resources is granted according to these policies.

3.2 Design Requirements

The goal of this dissertation is to design an effective LBAC system that meets the following requirements: (1) security; (2) convenience for users of the system; (3) robustness; (4) cost; and (5) supports access policies that grant or deny access to resources. These are each discussed in the following subsections.

3.2.1 Security

One of the most essential characteristics of a LBAC system is security. In general, access control is intrinsically about security and focuses on ensuring that only authorized users obtain access to requested resources. Security is perhaps more critical when dealing with sensitive data and/or resources. Ensuring a system's security includes having all the elements of the system protected from unauthorized access and shielded against potential attacks. Ensuring a system’s security includes securing the communications between the different nodes in the system by using secure links. A secure link is encrypted by one or more security protocols to ensure the security of data flowing between the nodes. Therefore, a secure system uses powerful encryption techniques. Furthermore, to ensure a system’s security, it is important to study the possible security vulnerabilities in the system, and then address them.

Another potential aspect of security is limiting access to resources when users are physically located inside an access area. Therefore, a user’s sessions should be terminated once outside the area defined for access.

3.2.2 User Convenience

User convenience is a very important factor for any system's success. User convenience can be indicated by the amount of burden that is put on users in order to perform authentication to obtain access to resources. It is well known that users are the weakest link in the security of systems. Therefore, systems that limit user interaction (or at least limit the physical interaction of users) fare better. The two-factor authentication system RSA SecurID requires users to input a six-digit code displayed on a physical token device, along with their username and password. Moreover, users must often enter a second code after a (sometimes) significant delay. Systems that require extra steps to authenticate users tend to drive them away. Often, such systems become stagnant because users seldom use them. Admittedly, user convenience is typically seen as a trade-off with respect to security [63]. However, is this really the case? We believe that, in many cases, it is possible to design a system that is convenient for users without compromising security. Moreover, this can be done by using available data that does not require extra user interaction to increase the system's security. An example of this is using cameras and face recognition technology to unlock an iPhone.

3.2.3 Robustness

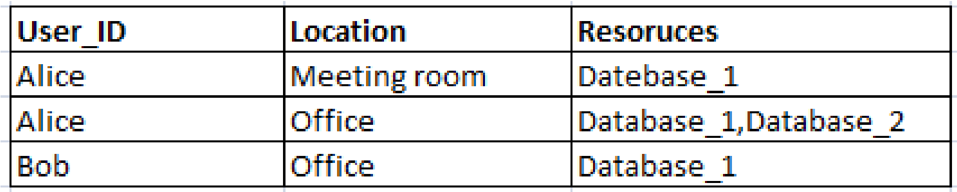
In computer science, robustness is defined as the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions [40]. In location-based systems, robustness is essential due to the unsteady nature of location data. Location data often fluctuate; therefore, the system must anticipate these kinds of changes and act accordingly. Furthermore, an efficient LBAC system must be optimizable so that tolerance and rigidity are assured in order to achieve the best results for a particular deployment.

3.2.4 Cost

The financial aspect is an important aspect when an organization is determining an appropriate LBAC system to use. Very sophisticated and accurate LBAC systems can be designed using customized hardware if money is not an issue. However, cost is important and must be minimized when possible. The cost of a system includes equipment costs, deployment costs, maintenance costs, and so on. It is important to consider the cost when designing a LBAC system. Usually, more complex systems are harder to install and maintain, resulting in higher costs. A cost-efficient LBAC system minimizes new equipment, installation, and maintenance costs.

3.2.5 Location-based Access Policies

A typical access control system in an organization usually supports multiple resources and multiple access rights and permissions. Traditional access control policies manage who accesses what. However, in LBAC systems, access control policies manage who accesses what – and where. It is essential that an LBAC system is configured with the desired access policies and subsequently follows these policies reliably. Figure 3-2 illustrates an example of a typical location-based access policy.



User_ID	Location	Resoruces
Alice	Meeting room	Datebase_1
Alice	Office	Database_1,Database_2
Bob	Office	Database_1

Figure 3-2: An example of a location-based access policy.

3.3 Proposed System Design

In this section, the proposed system's design is presented. By design, it must meet the design requirements discussed above. Moreover, the proposed design is set to overcome the weaknesses in other systems noted in the related work in the previous chapter. The proposed system utilizes IEEE 802.11 (Wi-Fi) technology, which helps meet various design requirements as it is inexpensive and broadly available. This results in a LBAC system that can be deployed right now in just about any setting.

The proposed method of localization utilizes broadcasted signals to provide an accurate and secure LBAC system. Access areas, which are areas where access to resources are granted, are defined by coverage of one or several access points. These access points broadcast a unique token every predefined period. This period can be varied (and is discussed later). A user's device collects the broadcasted tokens from the access points and their corresponding RSSI values and uses this information to authenticate to the authorization entity.

In the experimental environment in which the proposed system was implemented and tested, the authorization entity is software that is added to the resource (which is a simple server). It is in charge of authenticating and authorizing users for access to resources. In this text, authorization entity and the resource server are used interchangeably to reflect the experimental environment. In practice, however, they could be separate entities. The authorization entity makes the decision to grant or deny access to resources. Moreover, it determines what specific resources to grant the user access to based on the access policy in place.

The proposed LBAC system, as is typical in access control systems, performs identification, authentication, and authorization. As shown in Figure 3-3, the goal of identification is to identify a user requesting access to resources. In a LBAC system, this can be done by requiring a username and by only granting access to identified users in approved locations. The goal of authentication is to verify the identity of a user requesting access to resources. In the proposed LBAC system, authentication is done by having the user's device send the broadcasted tokens of nearby access points (along with a password – if two-factor authentication is desired). Consequently, the goal of authorization is to authorize access to resources. Here, the authorization entity verifies the credentials received from the user's device and makes the decision to authorize or deny access. Moreover, it determines what specific resources to grant the user access to.

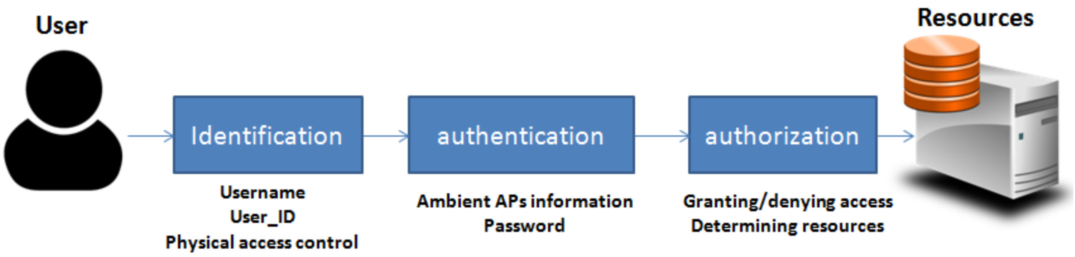


Figure 3-3: Identification, authentication and authorization in the proposed LBAC.

In the proposed system, the IEEE 802.11 protocol is utilized because it is cost-effective, extensively deployed, and already integrated into most of the infrastructure all over the world today. Furthermore, the beacon frame in the IEEE 802.11 protocol can be easily modified to carry tokens, and it can be broadcasted repeatedly over a very small period. The proposed design allows for existing or new IEEE 802.11 network interfaces to broadcast tokens for use in authentication. User devices can listen to and process tokens broadcasted by access points in range without necessarily being associated with or connected to them (e.g., for network access). The LBAC system allows users the freedom to choose their network connection using the access points they desire (e.g., any Wi-Fi network that can provide network access).

One of the most significant benefits of the proposed LBAC system is that it is a zero-interaction system (i.e., it is unobtrusive for users). This will provide convenience to users and will minimize the complexity of the system. Specifically, a user's device receives broadcasted tokens from the access points without needing to be connected to them (e.g., for network access). This is a passive process, which is different from existing methods. Access points can have multiple wireless network interfaces, typically corresponding to a unique Wi-Fi frequency (currently 2.4 GHz and 5 GHz for IEEE 802.11 Wi-Fi). However, some access points have more than one wireless network interface for each of the frequencies. Moreover, some access points support the creation and configuration of virtual network interfaces that can broadcast their own beacon frames. This is done by subdividing a physical interface into two or more virtual interfaces on which unique network parameters can be assigned (e.g., a physical address, an IP address, etc.). The benefit of this is that a specified network interface can be dedicated for token transmission to support the proposed system.

In the proposed LBAC system, the access area (i.e., the area where access to resources is to be granted) is defined by the coverage of one or multiple access points. As shown in Figure 3-4, an access area is best defined by the overlap of coverage of typically three token transmitting access points. The access area's size can be customized by defining appropriate RSSI values where access is to be granted and configuring the authorization entity with these values. Our design aims to grant access when the coverage of the token transmitting access points used for that access area overlaps (i.e., with appropriate RSSI values), and the user can collect the tokens broadcasted by these access points. More details about defining the appropriate RSSI values for an access area are discussed in Section 4.4.3 in the next chapter.

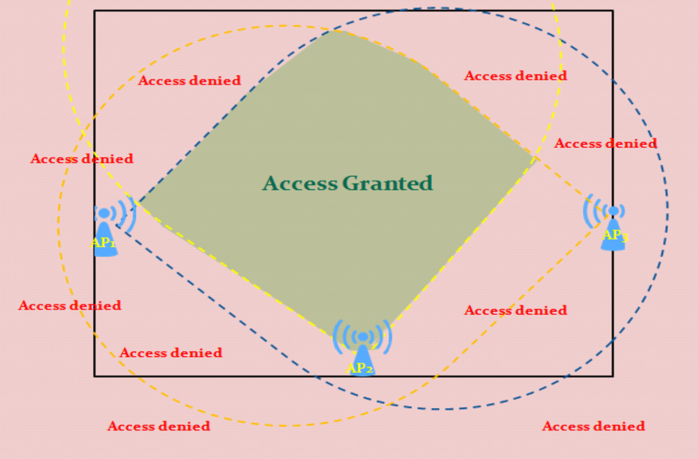


Figure 3-4: Defining an access area by access point coverage.

Most buildings today have IEEE 802.11 access points installed at different locations. They are mainly used to provide wireless connectivity to a network. The proposed system makes use of these access points (and their associated coverage) to define an area where access to resources is granted. As previously discussed, the RSSI values of the access points at the mobile device's location give an indication of the location of the mobile device in reference to these access points. For example, having a device in the middle between two access points is reflected by the RSSI values at the mobile device's location and indicates that the device is located between them. This is because the RSSI measurements for the two access points in this case will be close to each other. Therefore, in the proposed system, the RSSI values of the token transmitting access points at the device's location are utilized in making the decision of granting or denying access to resources.

In order to make sure that users have access to resources if they maintain their presence in the access area. The user's device must repeatedly send the latest tokens broadcasted by the access points and their corresponding RSSI values every predefined period. If the user's device fails to do so, the authorization entity should immediately terminate the user's session and deny further access to resources. This period is configured in the authorization entity and is optimized as it presents a trade-off between security and efficiency. This period must not be too long that the user can maintain access outside the access area, and not too short that it will consume resources at the access points, user's device, and the authorization entity.

In the proposed system, the authorization entity is in charge of verifying the credentials provided by the users. As previously discussed, in the proposed system, these credentials include the tokens broadcasted by the access points and collected by the user's device along with corresponding RSSI values. Consequently, the authorization entity is responsible for verifying these tokens, and if the RSSI values are within range of the ones configured at the authorization entity for a specific access area. Furthermore, the authorization entity is configured with the access policy that contains the users' profiles and their access rights at different locations.

So far, the proposed LBAC system consists of the authorization entity with the access policy at one end, and the access points that are modified to broadcast tokens periodically using the beacon frame at the other end. Users then employ the tokens and RSSI values of these access points to authenticate to the authorization entity. Figure 3-5 illustrates an overview of the proposed system's operation, and is followed by a step by step explanation:

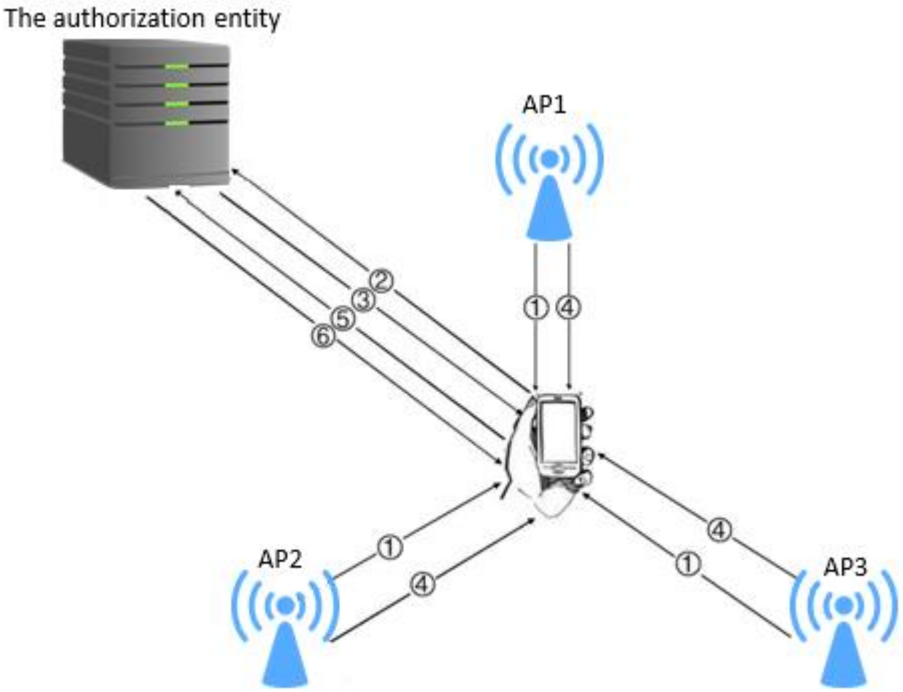


Figure 3-5: The proposed system operation.

1. At time $t1$, the access points (three in the figure above) broadcast tokens simultaneously using the beacon frame. The user's device collects the tokens, their

corresponding RSSI values, and the MAC addresses of the access points. The user's device then derives the location proof using the following formula:

$$LP(t) = ||_{i=0}^{n} MAC_{APi}.Tk_{APi}(t).RSSI_{APi}(t)$$

Eq. 3-1

Where

$LP(t)$ is the location proof at time t.

| | denotes concatenation.

n is the number of token transmitting access points in range of the user’s device.

MAC_{APi} is the MAC address of the access point APi transmitting the token.

Tk_{APi} is the deobfuscation of the obfuscated token broadcasted by the access point APi.

$RSSI_{APi}$ is the RSSI (in dBm) measured at the user's device for access point APi.

As can be seen from Equation 3-1, the location proof is the concatenation of the tokens transmitted by the access points, the RSSI measurements of these access points at the user's device, and the MAC addresses of the access points.

2. When the device is authenticating with the authorization entity, the user's device sends this location proof to the authorization entity.
3. The authorization entity receives the location proof. It subsequently verifies the tokens and determines if the RSSIs correspond to the configured values of an access area. These RSSI values only appear if the user is inside the specified access area. Therefore, if the location proof contains valid RSSI values, they will fall in the configured ranges of the access area at the authorization entity. This means that the

user is inside the access area. Then, access to resources is granted or denied based on validity of the tokens and the RSSIs.

4. As in step 1, at time $t2$, the access points simultaneously broadcast new tokens. The user's device collects the new tokens and their corresponding RSSI values and derives a new location proof $LP(t2)$ using Equation 3-1.
5. In order to maintain access to resources, the user's device then sends the new location proof to the authorization entity.
6. As in step 3, the authorization entity verifies the received location proof, and either continues to grant access to resources or terminates access to resources.

The proposed system is expected to meet the LBAC design requirements of cost as it utilizes inexpensive and ubiquitous IEEE 802.11 access points. The proposed design is expected to be secure as it requires the user's device to continue providing a new location proof to make sure that the user's device is still physically present in the access area.

The period for requiring new authentication information from devices should be carefully optimized and configured at the authorization entity. It should not be so small that it drains resources at the user's device (e.g., processing power, battery, etc.), nor should it drain the system's network resources (e.g., causing large overhead and slowing down data transmission). On the other hand, it should not be so large that it is possible for users to leave the access area without having their access to resources terminated.

The proposed method allows flexibility with respect to authentication, as it can be deployed with or without requiring a username (in this case, a temporary user ID can be assigned instead). It is possible to use the tokens broadcasted by the access points as the only authentication factor; however, they can be used as a second authentication factor along with a password.

In the proposed system, the token transmitting access points, as discussed later, generate tokens locally using pre-shared keys. A pre-shared key is a shared secret that was previously agreed upon between two parties. Thus, token generation is done locally on both parties (i.e., on the access points and the authorization entity) to minimize extra hardware and/or connections, and more importantly, doing so elsewhere could increase the risk of attackers intercepting this process and obtaining knowledge of the token generation process. Each token broadcasted by the access points should be unique at every point in time so that historical data cannot be used to determine a future token.

The RSSI values of the token transmitting access points are used in the authentication process because they provide an indication of the user's location in reference to the access points. Thus, our proposed system requires initial calibration at deployment to collect the range of RSSI values from token transmitting access points in the desired access area. The system uses this data to configure the authorization entity to aid in making resource authorization decisions. The calibration process helps to set precise access areas. This process can be performed manually by dedicating devices for this purpose or by crowdsourcing the problem and utilizing devices of trusted users to obtain RSSI values throughout a potential access area as in [11]. Crowdsourcing is a process through which a task is completed through a group of participants. In the proposed system, the calibration process can be done by using the RSSI values recorded at trusted users’ devices to configure an access area’s accepted ranges at the authorization entity.

One of the strengths of our proposed design is that it utilizes existing IEEE 802.11 infrastructure and builds on top of it by adding the LBAC functionality. In the next section, the potential challenges of achieving the proposed system are discussed. These are synchronization, security of tokens, and access areas.

3.4 Design Challenges

The proposed system is intended to meet the design requirements for an effective and efficient LBAC system. It consists of multiple elements that must be carefully designed in order to achieve the objective of this research. Below are some of the challenges in achieving such a system.

3.4.1 Synchronization

Synchronization is essential for security in LBAC systems, and it is a very important aspect of the proposed system. In our design, access points are modified to broadcast tokens using the beacon frame. Collectively, tokens (along with RSSI values) provide a location proof. The tokens are time-sensitive, in that a user must have the latest tokens from all participating access points in order to be successfully authenticated. Therefore, all participating access points in the system must be synchronized, along with the authorization entity. Synchronization allows users to generate valid location proofs and permits the authorization entity to successfully verify the location proofs of users requesting resources. If the system becomes inoperable, from a power outage, for example, the process of synchronization places the system back into an operative state.

3.4.2 Security of Tokens

To combat the potential threat of attackers attempting to gain unauthorized access to resources in the system by generating their own valid tokens (thereby bypassing the access points entirely), it is important to discuss the manner in which the tokens are generated. In our proposed system, the access points generate and transmit tokens that the user’s device utilizes to derive a location proof. Securing the process of generating and transmitting tokens can be done through the use of efficient cryptographic schemes that utilize a pre-shared secret. The authorization entity should be able to verify the legitimacy of the location proofs of users; therefore, it is essential to determine the best fitting type of encryption algorithm.

3.4.3 Access Areas

Access areas describe the areas where access to resources is granted so long as a user (meaning a user's device) is physically located within them. In the proposed system, access areas are implemented by cleverly utilizing RSSI values of participating access points. Specifically, the RSSI values are collected by user devices and provide a relative strength of the signal received from the access points. This can subsequently be used to infer a location or distance from the access points. By utilizing a collection of access points, a user's position can be accurately determined. Admittedly, an initial calibration must be performed so that RSSI values relative to each of the access points are known. It should be noted that there are edge cases; these are areas outside of the access area (where access to resources should not be granted) that the authorization entity may unwittingly consider within the access area. This is a result of the potential for error in the RSSI values (i.e., they are not extremely precise). This possibility must, of course, be minimized; after all, a good LBAC system should not grant access to resources outside of defined access area(s). It is therefore essential that the proposed LBAC system is configured correctly such that the system minimizes false positives and false negatives.

On the other hand, it is important to note that, different user devices will result in different RSSI values for the same access points. Therefore, it is not the actual RSSI values that are important; rather, it is the combination of all of the collected RSSI values that establishes a relative position that is specifically based on the user's device. Therefore, it is essential that the proposed LBAC system is configured correctly to address the differences in device capabilities.

3.5 Conclusion

LBAC systems control access to resources utilizing location information via location proofs. Moreover, they are essential in today's mobile environment, where virtually everything is accessible through a computer network. People use LBAC systems in everyday life for different functions (e.g., making payments, in banking applications, merely accessing information, etc.). We strongly believe that strengthening the security of LBAC systems is research-worthy. Moreover, structuring the system in such a way that minimizes user interaction increases security, user-convenience, and robustness.

In this chapter, design requirements for an effective LBAC system were presented. Subsequently, we proposed a new LBAC system design that is intended to meet the proposed design requirements. Challenges involved in achieving such a system were then discussed. Achieving the proposed system with good performance is significant, because it would provide a LBAC system in which the location of the user is continuously verified in a way that guarantees authorized access to resources when users are physically present in predefined access areas. Our proposed LBAC system does this in a convenient way precluding effort from users, and without requiring additional infrastructure. This is the motivation for utilizing ubiquitous IEEE 802.11 infrastructure, and will be discussed in the next chapter, by simply modifying the firmware of existing access points. We believe that designing an unobtrusive, inexpensive, and secure LBAC system will encourage an increase in the integration of LBS to access control systems in the future.

CHAPTER 4

IMPLEMENTATION

In this chapter, we discuss implementation specifics that takes our design (from the previous chapter) and implements it in a research testing environment. In addition, several details omitted in the design are included here for completeness. Section 4.1 examines the deployment overview, Section 4.2 addresses the design requirements in an applied manner, Section 4.3 enumerates the system's components and their implementation specifics, Section 4.4 presents the experiments to test our proposed design, Section 4.5 discusses an example scenario, and we conclude in Section 4.6.

4.1 Deployment Overview

Due to the added task of determining user locations, a LBAC system typically has more elements than a regular access control system. The proposed system, as shown in Figure 4-1, consists of the following three elements:

1. *The user's device with custom designed client software installed.* The client software is installed on the user's device and enables it to integrate within the proposed LBAC system. More details about the client software will be discussed in detail later in this chapter.
2. *The access points that broadcast the tokens used as location proofs*. In general, the number of access points used for authentication depends on the specifics of

deployment. Characteristics such as, for example, the desired security level, the building's layout, obstacles and materials used in the building, and so on will affect this. The proposed system allows implementing access areas using a single access point. However, three access points at minimum are required to determine localization sufficiently (via triangulation). In more complex environments (or in environments with more complex access areas), however, additional access points may be needed.

3. *The authorization entity that verifies a user's location proof and makes the decision to grant or deny access to resources*. It additionally determines the resources that are granted to the user according to the access policy in place.

As seen in Figure 4-1, the direction of the arrows denotes the direction of communication. A user's device does not need to establish communication with participating access points. It passively detects the access points by scanning for and obtaining their broadcasts and for RSSI values and inspecting the beacon frame for tokens – to derive a location proof. However, a user's device must establish two-way communication with the authorization entity in order to get authenticated, and then authorized to get access to resources.

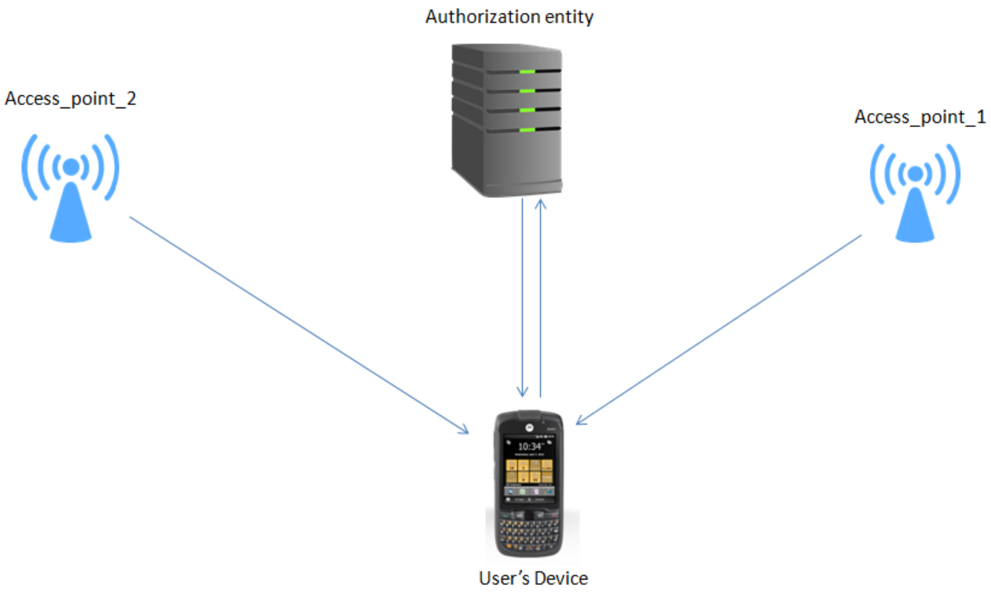


Figure 4-1: The proposed system's overview.

As previously discussed, the number of participating access points depends on deployment specifics and is further configured at the authorization entity. Since the proposed system uses existing IEEE 802.11 infrastructure, defined access areas depend on the position of access points in a location (e.g., a building). Moreover, the decision of access area locations can be made by taking an inventory of existing IEEE 802.11 access points and including some or all of them in the proposed LBAC system. Figure 4-2 illustrates an example of a deployment in a generic office setting. As seen in the figure, the locations of existing access points in the building are used to determine the access areas. If, for example, an access area is desired in offices 1, 2, and the conference room, access points AP2 and AP3 can be used to obtain tokens and RSSI values. Similarly, an access area for offices 3 and 4 can be created using token and RSSI values obtained from access points AP1 and AP3. An access area for the cubicles (in the center of the building space) can be created using access points AP3, AP4, and AP5. Of course, the proposed

LBAC system supports the addition of new access points that can be precisely positioned to more precisely define access areas if desired.

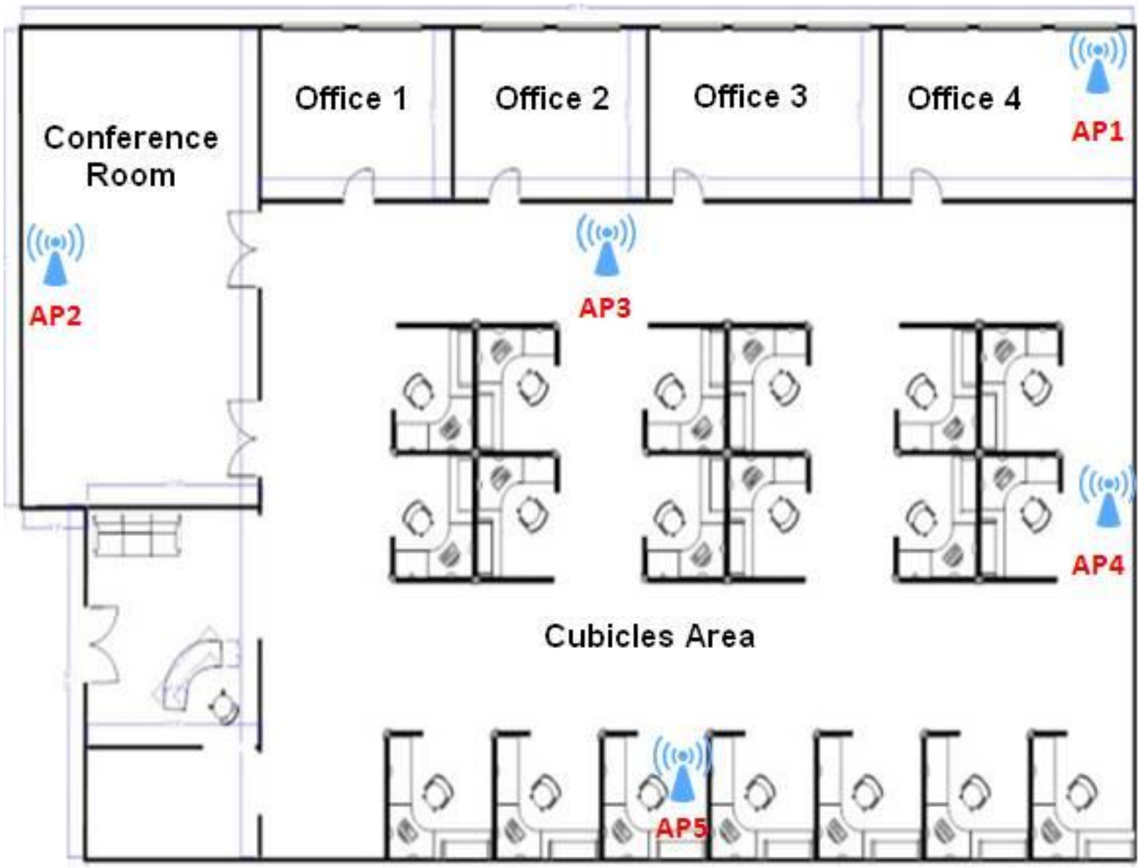


Figure 4-2: An example for the access areas configuration based on the floor plan.

4.2 Design Requirements

In order to successfully build the proposed system, the different elements of the system should be designed carefully so that they perform their tasks securely, effectively, and robustly. In this section, the design requirements for each of the system's elements are discussed. Specifically, the elements include the user's device (e.g., a smartphone), the participating access points (that generate the tokens), and the authorization entity.

4.2.1 Access Points

In the proposed system, access points are tasked with broadcasting the tokens used by the user devices to ultimately derive a location proof. This section discusses the design requirements for the access points.

4.2.1.1 *Configurability*

The proposed design relies on having the access points broadcast tokens using the IEEE 802.11 beacon frame. Specifically, we utilize the SSID field. Regular off-the-shelf (OTS) access points do not typically permit programmable (dynamic) modification of the beacon frame, including the SSID field. To support this, open access points must be obtained, or existing access points must be flashed with a firmware that supports dynamic modification of the beacon frame. Furthermore, the firmware must support scheduling the task of repeatedly broadcasting tokens.

4.2.1.2 *Broadcast period*

The access points in the proposed system are intended to broadcast new tokens every fixed period. The access points must be able to perform this function in a reliable manner, minimizing the time that an access point needs to change the beacon frame and broadcast it.

4.2.1.3 *Access point functionality*

The proposed system utilizes the existing IEEE 802.11 infrastructure but requires a modification of the firmware in participating access points. Moreover, custom software that generates tokens and modifies the beacon frame must be placed on participating access points before they can work within the proposed system. Consideration must be made to ensure that modifications do not compromise other functionality in the participating access points (e.g., providing access to a network).

4.2.1.4 *Processing power*

Participating access points must have sufficient processing power to perform the mathematical operations related to the cryptographic functions that are necessary to generate the tokens. This includes having enough processing power and memory to execute the custom software that generates the tokens without affecting the access point's normal functionality (e.g., to provide access to a network). In our initial experiments, ASUS RT-N12 access points were utilized which have a 300 MHz processor and 32 MB RAM. Unfortunately, performance was not acceptable; therefore, access points with an 800 MHz dual-core processor and 256 MB RAM were utilized instead. Details of the access points are discussed later in this chapter.

4.2.2 User Devices

The device of users who wish to use the system must support IEEE 802.11 functionalities (i.e., the ability to scan for Wi-Fi access points). The proposed system supports a wide range of devices whose hardware and software requirements are available in most of the mobile devices sold in the market today (e.g., smartphones, laptops, and tablets).

4.2.3 Authorization Entity

As previously stated, the authorization entity is tasked with verifying a user's credentials, deciding whether to grant access to resources, and determining the resources that a user has access to. This section discusses the authorization entity's design requirements.

4.2.3.1 *Connectivity*

The authorization entity must support connectivity using common network protocols. In our experiments, the SSH protocol is used for the connectivity between a user's device and the authorization entity. Although the SSH protocol was used to test connectivity of the users' devices to the authorization entity, it should be noted that any connection protocol can be used (e.g., web-based server, file transfer protocol (FTP), etc.). There are many different connection protocols, and choosing which depends on the resources available.

4.2.3.2 *Processing power*

The authorization entity is tasked with authenticating users (based on location and tokens obtained from participating access points) and making access control decisions for all users using the LBAC system. Thus, it must have enough processing power to support all anticipated users. In general, the processing power required by the authorization entity will depend on the number of users in the system.

4.2.3.3 *Robustness*

In the proposed system, the authorization entity requires the most processing power and network bandwidth. Moreover, it is essential to determine who gets access to what resources. Therefore, it carries the greatest responsibility in the system. A failure in the authorization entity could render the entire system inoperable. Of course, this is the case with any resource entity implementing an access policy in any current system. It is therefore paramount that recovery plans be in place in case of failure. Details about restoring synchronization are discussed later in this chapter.

4.2.3.4 *Permissions and access rights support*

The proposed system serves a variety of users with different roles and different access rights. Therefore, selecting an appropriate operating system and file system to support this is important. This is discussed in more detail later in this chapter.

4.3 Design Specifics

In the previous section, design requirements for each of the system's elements were introduced. This section covers hardware and software specifics of the system that meet the design requirements. Moreover, they reflect the specifications used in our experiments. Section 4.3.1 discusses the design specifics of the access points, Section 4.3.2 discusses the design specifics of user devices, and Section 4.3.3 discusses the design specifics of the authorization entity. Later in this chapter, the operations of these elements are clarified with an example scenario.

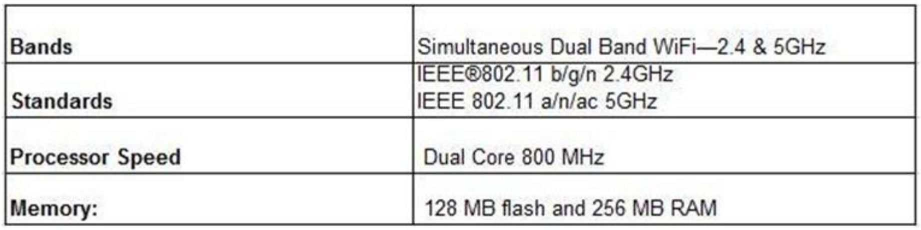
4.3.1 Access Points

In the proposed system, the Wi-Fi access points perform the usual access point function, which is providing access to a network. Furthermore, they broadcast the tokens that are used for the LBAC system. Since this is done in the SSID field of the beacon frame, the access points must be configurable and programmable. In most cases, the default firmware on OTS access points is not able to perform the necessary tasks required by our proposed system; therefore, third-party firmwares (such as DD-WRT, OpenWrt, Tomato, etc.) must be used and flashed on the access points. For our experiments, the DD-WRT firmware [see 17] was utilized.

DD-WRT is a Linux-based third-party firmware for wireless routers and access points that supports a wide variety of access point models. In general, DD-WRT is one of the few firmwares available that offers a powerful set of additional features and functionalities to provide more programmability and flexibility to the access points. There are various versions of DD-WRT, each of which supports different access point models. For our experiments, we used DD-WRT version 3.0-r37015M Kongac on the Netgear R6400 Wi-Fi access point.

For our experiments, we controlled the participating access points via the Linux command line (or terminal), which is supported through the DD-WRT firmware. Most importantly, we needed to control the access point’s wireless network interface for token transmission. Moreover, several underlying applications were installed to support additional functionality: SSH was required for remote management of the access point; the Bash shell was required to create and execute software (in the form of scripts), and OpenSSL (a library for implementing cryptographic functions) was required to generate tokens properly.

After researching different models of access points for support of these functionalities, the Netgear R6400 was selected. It is a dual-band access point (i.e., it operates on both 2.4 GHz and 5 GHz frequencies). This allows it to be used in the LBAC system while simultaneously providing network access. Figure 4-3 shows the Netgear R6400 access point and its hardware specifications.



Bands	Simultaneous Dual Band WiFi—2.4 & 5GHz
Standards	IEEE®802.11 b/g/n 2.4GHz IEEE 802.11 a/n/ac 5GHz
Processor Speed	Dual Core 800 MHz
Memory:	128 MB flash and 256 MB RAM

Figure 4-3: The Netgear R6400 and its hardware specifications.

As previously mentioned, participating access points modify the beacon frame and use the SSID field to broadcast tokens. In order to maintain the functionality of the access points to provide access to a production network, either one of the bands (2.4 GHz or 5 GHz) can be used for token transmission. In general, each of the bands' physical interfaces can be used to broadcast tokens. However, some access points allow configuring a virtual network interface that can be used to transmit tokens. A virtual interface provides a virtual version of a physical device that, through software, maps the virtual device to the physical device and utilizes the operating system's underlying device drivers. In our experiments, a virtual interface was created and configured to broadcast the tokens. As shown later in this chapter, we experimented broadcasting tokens using both frequencies.

4.3.1.1 *Token generation*

As previously mentioned, the tokens are generated by utilizing a pre-shared secret between the access points and the authorization entity. In our proposed LBAC system, tokens are generated using the HMAC function as shown in Figure 4-4 and as follows:

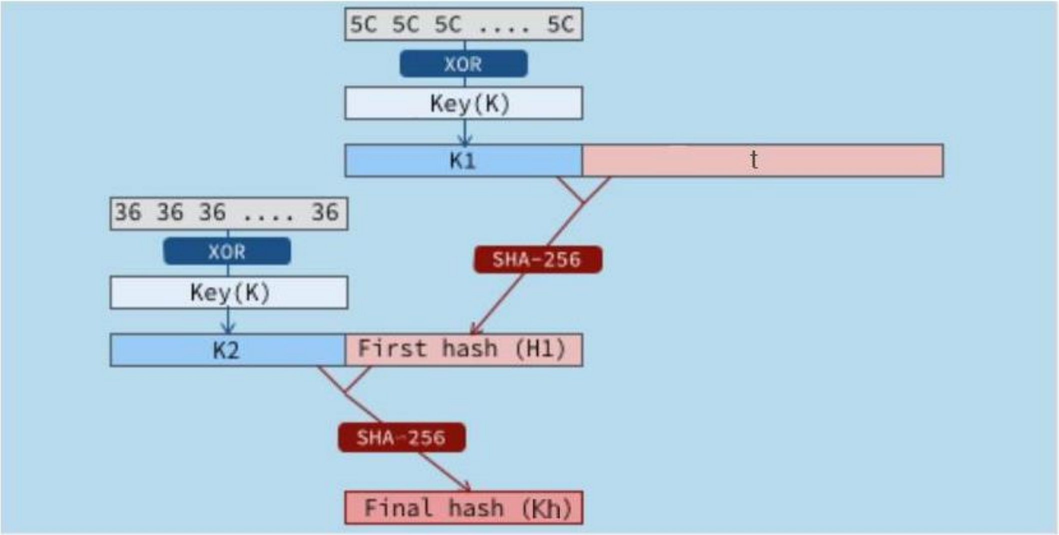


Figure 4-4: The HMAC-SHA-256 code generation process [19].

$$K_h = HMAC(k,t) = H(\,(k\, \oplus\, i)\,||\,H((k \oplus o)\,||\,t)) \tag*{\textbf{Eq. 4-1}}$$

Where

H is a cryptographic hash function. Secure Hash Algorithm-256 bits (SHA256) is used in the proposed system, which is a cryptographic hash function developed by the United States National Security Agency. More details about selecting SHA256 are discussed in the next chapter.

t is a timestamp in the format YYYYMMDDHHmm, where YYYY is the 4-digit year, MM is the two-digit month, DD is the two-digit day of the month, HH is the two-digit hour (in 24-hour format), and mm is the two-digit minute.

k is the pre-shared secret, which is in ASCII (American Standard Code for Information Interchange) text form. ASCII is a character encoding standard for electronic communication.

$\|$ denotes concatenation.

$\oplus$ denotes bitwise exclusive or (XOR), which is a binary operation.

o is the outer padding, consisting of the repeated hexadecimal bytes 0x5c (up to the block size). Padding is the process of filling a field with pad characters, which in this case are the repeated hexadecimal bytes 0x5c.

i is the inner padding, consisting of the repeated hexadecimal bytes 0x36 (up to the block size).

In this research, two possible scenarios were considered: one where the access points have access to a time server for synchronization, and another where they do not. The description of these scenarios is as follows:

4.3.1.1.1 Case 1: The access points have access to a time server

In this case, the access points and the authorization entity have access to a time server. A time server is a separate server on the network that has access to the exact time and distributes this information to clients that request it. In general, when two nodes synchronize their time with a time server that is synchronized with actual time, they are synchronized. Figure 4-5 illustrates the token generation process that uses a synchronized time as input:

1. Using the timestamp and the pre-shared secret, the HMAC function generates a ciphertext using Equation 4-1. This function outputs the code K_h, which consists of 32 bytes.

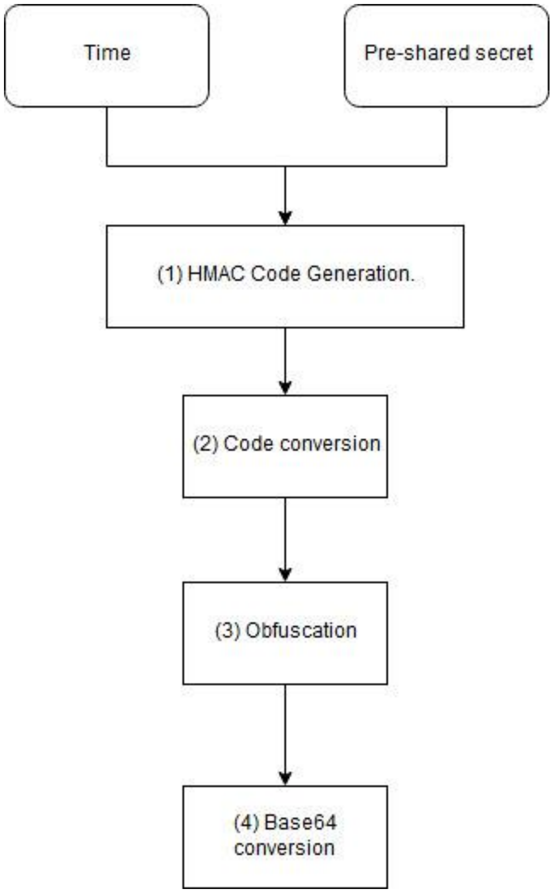


Figure 4-5: The token generation process in case the access point has access to a time server.

2. Due to the limited number of characters in the SSID field in the beacon frame, a 6-digit decimal key is generated from the HMAC function as follows:

$$T = K_{h} \, mod \; 1000000 \qquad \textbf{Eq. 4-2}$$

   Where mod denotes the modulo operator, which finds the remainder of the division. The output T consists of a 6-digit decimal key, which is what a user's device ultimately uses for authentication.

3. The token is translated via shifting and multiplying by constants. Translation is performed as follows:

$$Ts = a * T + b \qquad \textbf{Eq. 4-3}$$

Where T is the token derived at step 2,a and b are 6-digit integers. The length of a and b are constrained by the size limit of the SSID field (32 characters). The length of Ts varies depending on a, T, and B. This step helps mitigate attackers who do not have client software installed on their device. Without knowledge of the translation constants, attackers will not be able to decipher token values; consequently, attackers will not be able to generate verifiable location proofs. More details about this are discussed in the next chapter.

4. The translated token is encoded to base 64, generating Tb. This final token is ultimately broadcasted by the access point along with the header, which consists of fixed characters, so the user's device can identify token broadcasts. Base 64 encoding output size depends on the input size; it merely assures that whatever is encoded can be represented entirely with printable characters (a requirement of the SSID field).

4.3.1.1.2 Case 2: The access points do not have access to a time server

In this case, the access points have no access to a time server. Thus, a counter is used to generate the tokens. A counter is started at the exact same time in the participating access points and the authorization entity, which is done by initially synchronizing the access point and the authorization entity when setting up the system before starting a counter. The counters ensure that the participating access points are in synchronization with the authorization entity. More details about restoring synchronization if faults occur are discussed later in this chapter. Figure 4-6 illustrates the token generation process in the absence of a time server:

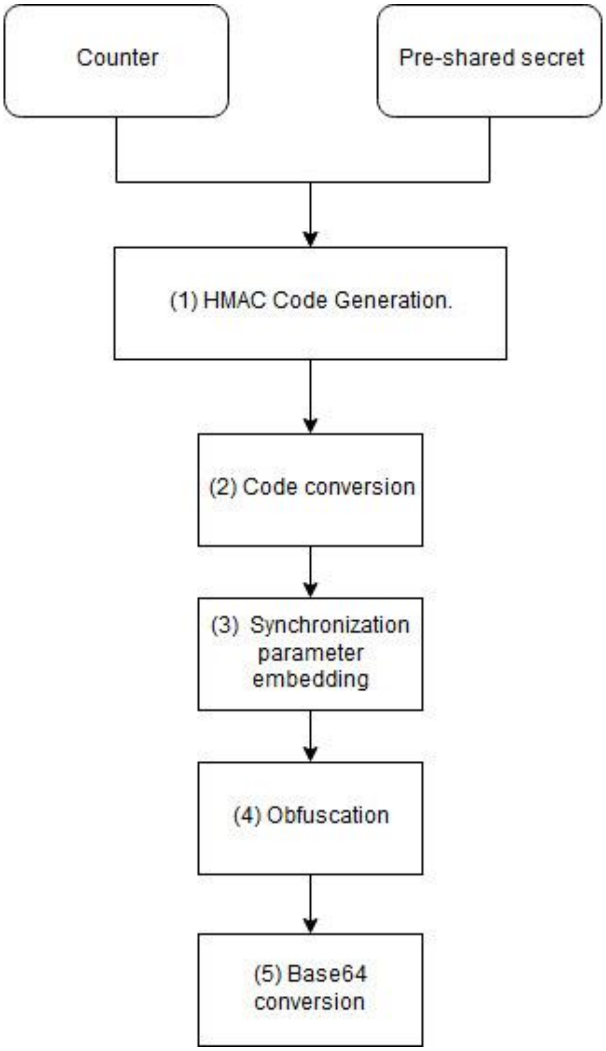


Figure 4-6: The token generation process in case the access points have no access to a time server.

1. Using the counter and the pre-shared secret, the HMAC function generates a code using the following equation:

$$K_h = HMAC(k,c) = H(\ (k\ \oplus\ i)\ ||\ H((k \oplus o)\ ||\ c)) \qquad \textbf{Eq. 4-4}$$

Where

H is a cryptographic hash function. The SHA256 hash function is used in the proposed system.

c is the counter value, which is obtained from the local clock of the access point in the format of YYYYMMDDHHmm, where YYYY is the 4-digit year, MM is the two-digit month, DD is the two-digit day of the month, HH is the two-digit hour (in 24-hour format), and mm is the two-digit minute. An example of a counter value can be “201903251109”.

k is the pre-shared secret, which is in ASCII text form.

‖ denotes concatenation.

$\bigoplus$ denotes bitwise exclusive or (XOR).

o is the outer padding, consisting of the repeated bytes 0x5c (up to the block size).

i is the inner padding, consisting of the repeated bytes 0x36 (up to the block size).

This function outputs a code K_h, that consists of 32 bytes.

2. Similarly, a 6-digit decimal key is generated from the HMAC function output following Equation 4-2. The output T consists of a 6-digit decimal key, which is what the user ultimately uses for authentication.
3. The token T is concatenated with the synchronization parameter, which is the local time in DDHHmmSS format, where DD is the two-digit day of the month, HH is the two-digit hour (in 24-hour format), mm is the two-digit minute, SS is the two-digit second:

$$T_c = T||S \qquad \textbf{Eq. 4-5}$$

Where T_c is the token after adding the counter information, T is the token generated in Step 3, S is the synchronization parameter in DDHHmmSS format, which is used to restore synchronization with the authorization entity.

4. $T_s$is translated via shifting by a constant and multiplying by another constant following Equation 4-3.
5. The translated token is encoded in base 64, generating Tb. This final token is broadcasted by the access point along with the header, which consists of fixed characters, so the user’s device can identify token's broadcasts.

The access point transmits a base 64 representation of the token after the SSID header. The SSID header consists of three fixed ASCII characters, and it is used to declare that this Wi-Fi interface is used for broadcasting tokens; consequently, the user's device scans and identifies this interface using the SSID header, and then extracts tokens from these broadcasts. The access points transmit this information utilizing the SSID field of the beacon frame. The DD-WRT access point firmware allows modification of the SSID field. For our experiments, custom software was created to modify the SSID field to include the token. This process involves: (1) generating the token; (2) replacing the SSID field in the beacon frame with the token; and (3) restarting the interface to apply the new beacon information. This process is done repeatedly by the access point every predefined period. This period is configured in participating access points and the authorization entity and is optimized in a way that fits the deployment the best. More details about this are discussed in the next chapter.

4.3.1.2 *Access areas*

In the proposed system, access areas are areas where access to resources is granted. In general, access areas are defined by the coverage of the token transmitting access point with specific RSSI ranges, which are configured in the authorization entity. Figure 4-7 shows the projected shapes of access areas using one, two, and three token transmitting access points.

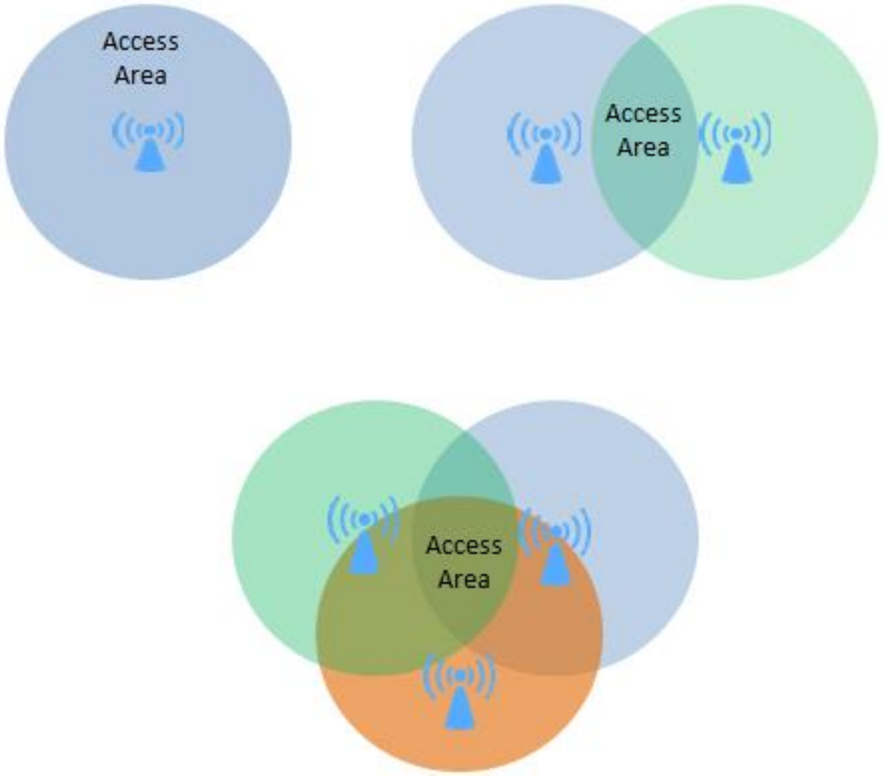


Figure 4-7: Access areas shape using one, two, and three token transmitting access points.

An access area's size can be modified using the accepted RSSI ranges of the access points in an access area. Since the vast majority of user devices in a typical application setting utilize very similar antennas (with similar power), RSSI ranges are acceptable to denote access areas. As shown in Figure 4-8, an access area's size can be increased by accepting a wider range of RSSI values at the authorization entity.

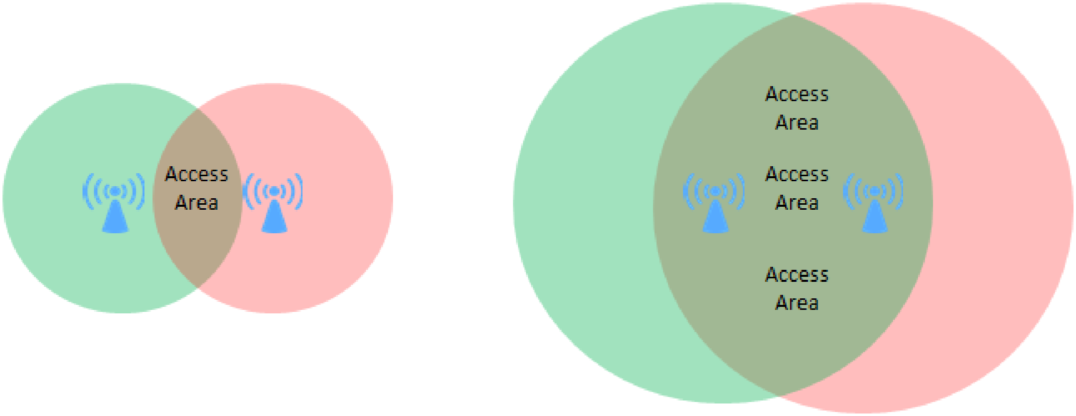


Figure 4-8: Access area size can be modified when using different RSSI ranges.

The proposed system aims to use the current IEEE 802.11 infrastructure without adding any new hardware. This is one of its strengths. Thus, it allows using any number of access points to define an access area. However, in order to achieve precise access areas, it is recommended to use at least three access points to achieve localization as in [49], for example.

The RSSI values in which access is granted in an access area can be collected manually, which is done by recording the RSSI values transmitted in the beacons of participating access points in each access area. This is clearly a manual (and likely time consuming) process; however, it is a one-time process. The same result can be achieved automatically, however, through crowdsourcing RSSI values in an area, which is done by utilizing the devices of trusted users to obtain RSSI values throughout a potential access area and using them to configure an access area’s accepted ranges at the authorization entity.

After the RSSI values are collected in an access area, the ranges of accepted RSSIs for each area are sent to the authorization entity and subsequently used to configure the system.

4.3.2 The User's Device

In the proposed system, the user's device passively detects IEEE 802.11 beacon frames in order to extract the tokens stored in the SSID field to use as a location proof in the LBAC system. This is done by using custom designed client software that is installed on the user’s device. It also establishes a connection to the authorization entity to obtain access to requested resources. In our experiments, the client software is implemented using the Python programming language on a Raspberry Pi and uses the SSH protocol to communicate with the authorization entity. The client software is pre-configured with information related to the specifics of deployment. It must be distributed to users of the system (as is typical with LBS applications). The client software is pre-configured with the following:

1. The SSID header, which is used to identify the token transmitting access points. In this research, three ASCII characters in the SSID field are used to declare that a Wi-Fi interface is used for token transmitting.
2. The shifting and multiplication constants for each of the token transmitting access points in the system. These are necessary to rule out attackers who do not have access to the client software and attempt to connect to the authorization entity. More details about this are discussed in the next chapter.
3. The authorization entity information.

Below are the functions performed by the user's device in the proposed system:

4.3.2.1 *Obtaining a location proof*

The client software is preconfigured with the SSID header that is used to identify the access points that are broadcasting the tokens that are needed for creating the location proof. For each of the access points, the user's device extracts the broadcasted tokens as shown in Figure 4-9:

1. When the user is inside an access area, the client software imports the pre-configured parameters for the deployment (which includes the SSID header, and the shifting and multiplication constants).
2. The user's device scans for IEEE 802.11 access points in range. It then extracts the SSID, the MAC address, and the RSSI values of the relevant access points.
3. The user's device decodes the base 64 token from the SSID field, yielding Ts.
4. The user's device applies the proper shifting and multiplication constants to retrieve the token T as follows:

$$T = \frac{Ts - b}{a} \qquad \textbf{Eq. 4-6}$$

Where Ts is the translated token, and a and b are 6-digit integers.

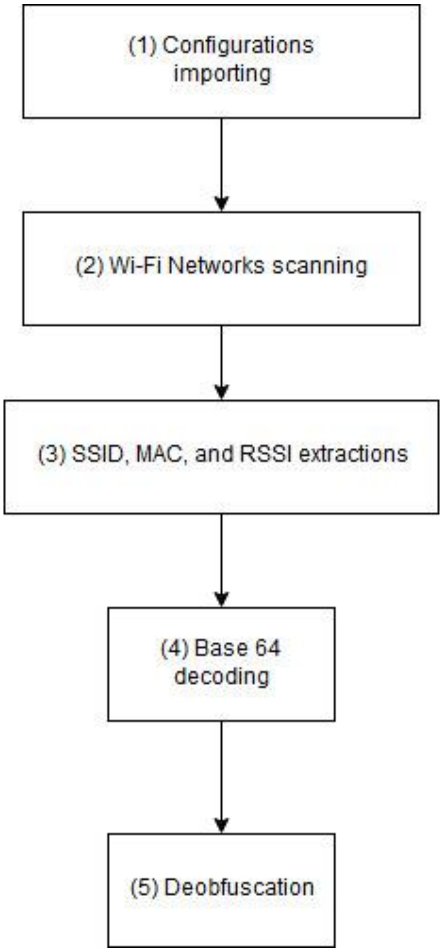

(1) Configurations importing
(2) Wi-Fi Networks scanning
(3) SSID, MAC, and RSSI extractions
(4) Base 64 decoding
(5) Deobfuscation

Figure 4-9: The process of extracting the tokens from access points broadcasts by the user's device.

In case a counter is used instead of the timestamp as in 4.1.1.12, then Tc is obtained as follows:

$$Tc = \frac{Ts - b}{a} \qquad \textbf{Eq. 4-7}$$

Where Ts is the translated token, and a and b are 6-digit integers.

5. The mobile device creates the location proof using equation 3-1. In case a counter is used instead of timestamp, the location proof is derived as follows:

$$LP(c) = ||_{i=0}^{n} MAC_{APi}.Tk_{APi}(c).RSSI_{APi}(c).c_{APi}(c) \qquad \textbf{Eq. 4-8}$$

Where

$LP(c)$ is the location proof derived using the value of the counter c.

| | denotes concatenation.

n is the number of access points used for a specific access area.

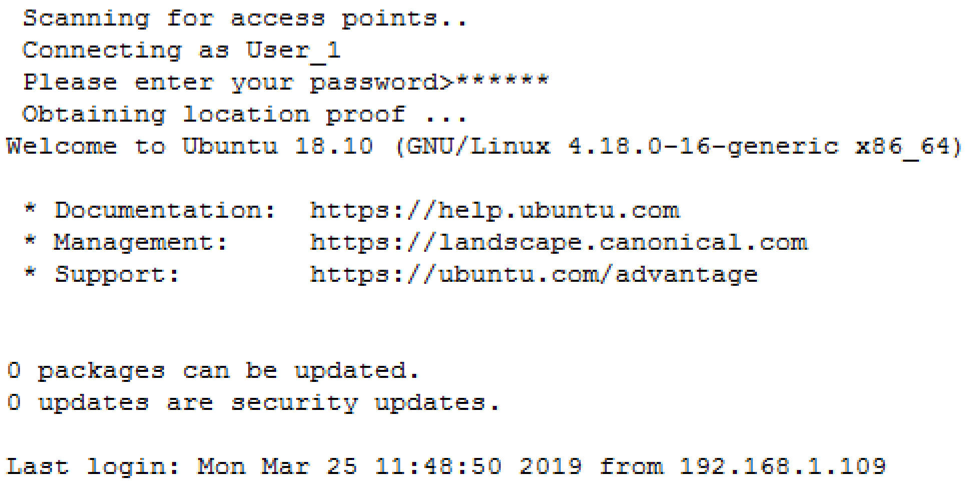
Tk_{APi} is the deobfuscation of the obfuscated token broadcasted by the access point APi.

$RSSI_{APi}$ is the RSSIs measured at the user's device for access point APi.

c_{APi} is the counter value broadcasted by access point APi.

4.3.2.2 *Connecting to the authorization (and obtaining access to a resource)*

The custom LBAC client software installed on the user's device automatically connects to the authorization entity in order to authenticate and subsequently obtain access to resources. The proposed system can be configured to use single-factor authentication (i.e., just a location proof) or multiple-factor authentication (e.g., a password along with the location proof). Adding a password as an extra authentication factor does not impact the performance since the user is prompted for the password only once when the SSH session is initiated; however, it adds an extra layer of security [54]. In our experiments, the SSH protocol is used to connect to the authorization entity. Figure 4-10 illustrates two-factor authentication on the client software at the user's device:



```
 Scanning for access points..
 Connecting as User_1
 Please enter your password>******
 Obtaining location proof ...
Welcome to Ubuntu 18.10 (GNU/Linux 4.18.0-16-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


0 packages can be updated.
0 updates are security updates.

Last login: Mon Mar 25 11:48:50 2019 from 192.168.1.109
```

Figure 4-10: The client software operation.

As can be seen in the figure, the location proof is automatically obtained from nearby access point scans. After access is granted, the user's device continues to scan for tokens and generates location proofs. These are continually sent to the authorization entity in order to maintain access to requested resources. This can effectively result in continuous authentication. The frequency in which the user's device sends the new location proofs is discussed in the next chapter.

4.3.3 The Authorization Entity

The authorization entity is tasked with identifying the user's access area, verifying user credentials, and applicably authorizing access to resources. Section 4.3.3.1 covers the token generation process at the authorization entity, Section 4.3.3.2 explains the process of identifying the user's access area, Section 4.3.3.3 covers authentication factor verification and authorizing access to resources, Section 4.3.3.4 discusses synchronization, Section 4.3.3.5 discusses continuous authentication, and Section 4.3.3.6 discusses access policy implementation.

4.3.3.1 *Token generation*

As discussed in previous sections, the authorization entity and the participating access points share a pre-shared secret that they use to derive tokens simultaneously and periodically. The authorization entity generates all of the participating access point tokens locally in the two cases discussed earlier (i.e., when the access points have access to a time server and when they use a counter for synchronization). Below are the details for token generation at the authorization entity for the two cases:

4.3.3.1.1 Case 1: The access points have access to a time server

In this case, the access points and the authorization entity have access to a time server, and they use it to synchronize time. Figure 4-11 shows the token generation process, which is as follows:

1. Using a timestamp and the pre-shared secret, the HMAC function generates a code following Equation 4-1. This equation outputs the code K_h, which consists of 32 bytes.
2. Similar to the access points, the authorization entity generates a 6-digit decimal key from the HMAC function following Equation 4-2.

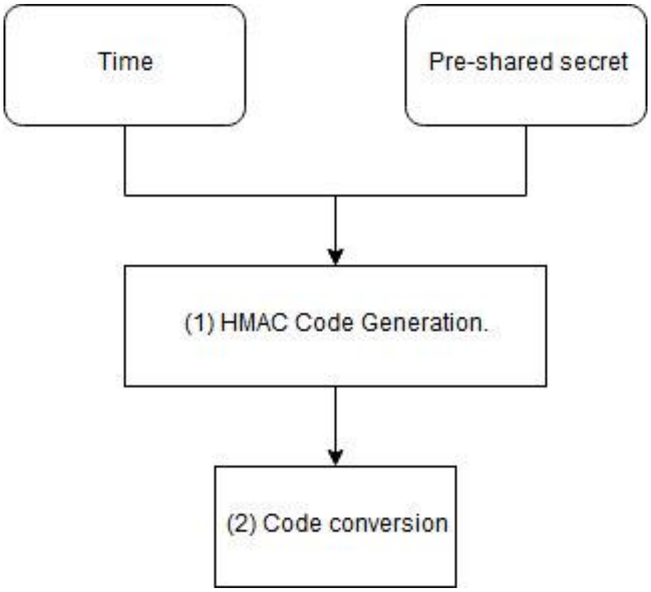


Figure 4-11: The token generation process at the authorization entity in the case the that access points have access to a time server.

4.3.3.1.2 Case 2: The access points do not have access to a time server

In this case, the access points have no access to a time server. Thus, they use a counter to generate the tokens. As previously discussed, the counter at the access points is initially synchronized with the one at the authorization. The counter at the access points must be always synchronized with the one at the authorization entity. This is done by utilizing the synchronization parameter broadcasted by the access points. More details about maintaining synchronization between the access points and the authorization entity is discussed later in this chapter. Figure 4-12 shows the token generation process, which is as follows:

1. Using the counter and the pre-shared secret, the HMAC function generates a code following Equation 4-4. This equation outputs the code K_h, which consists of 32 bytes.

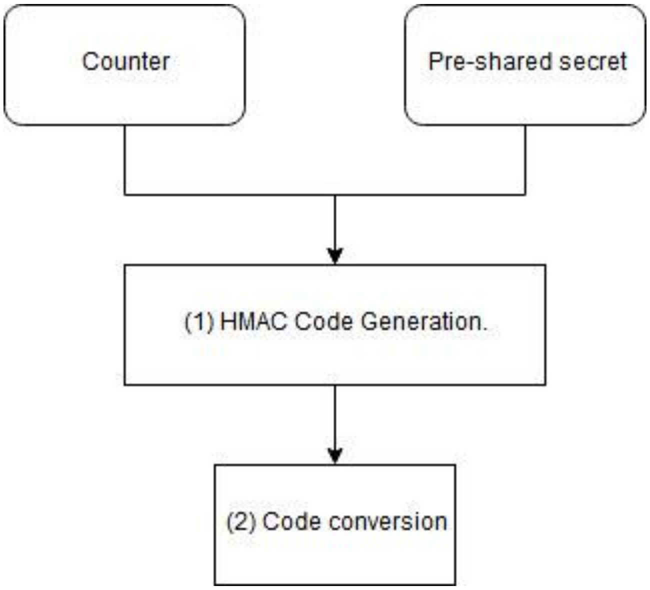


Figure 4-12: The token generation process at the authorization entity in the case that the access points do not have access to a time server.

2. The authorization entity generates a 6-digit decimal key from the HMAC function following Equation 4-2.

4.3.3.2 *Access area identification*

In the proposed system, the access area selection process is done automatically by the authorization entity. As previously mentioned, the client software is configured with SSID header of the token transmitting access points, and it scans and sends all relevant access point information to the authorization entity. As shown in Equation 3-1, the authorization entity determines the access area as follows:

1. The authorization entity compares MAC addresses of the token transmitting access points for each access area with the MAC addresses scanned and sent by the user's device. This returns the possible access areas.
2. As previously discussed, the accepted RSSI ranges of each access point in all access areas are stored at the authorization entity. Consequently, by looping through the

possible access areas returned from Step 1, the authorization entity checks if the received RSSIs from the user’s devices fall in the accepted RSSI ranges of any of these access areas. This is done by comparing the RSSI received from the user’s device for each access point in a specified access area with the RSSI range stored at the authorization entity for each access point in same access area. Then, if all the RSSIs received from the user’s devices fall in the ranges of the ones stored at the authorization entity, the access area is selected.

3. If no matching access area is found, authentication fails.

Access area selection is the first step of authentication. A failure in determining a user's location results in authentication failure. Furthermore, access area selection is essential for the verification of the location proof and authorizing access to resources as discussed in the next subsection.

4.3.3.3 *The verification of the location proof*

In our experiments, the SSH protocol was utilized for communication between a user's device and the authorization entity. Therefore, authentication was performed utilizing SSH. Most Linux operating systems support Pluggable Authentication Modules (PAM) that provide dynamic authentication support for applications and services in a Linux environment. PAM allows configuring the number and the nature of authentication factors. In our experiments, control of SSH authentication is transferred to the PAM library and is configured to perform two-factor authentication (a password and the location proof).

Location proof verification is done by linking the PAM configuration file to the token generation script (discussed previously) when an access request is received from a user. The user's credentials are subsequently verified: the user's password is first verified; then, the location proof sent by the user's device is compared to the one generated locally. Access to resources is granted when a valid password and location proof are presented by the user. In our experiments, the resources are modeled by an SSH session.

4.3.3.4 *Synchronization*

Because the proposed system uses time-based tokens, it is essential to maintain synchronization between the access points and the authorization entity so that they can generate the same time-based token. This subsection covers synchronization.

4.3.3.4.1 When the access points have access to a time server

In this case, the participating access points and the authorization entity have access to a time server; therefore, they are all synchronized. However, it is important to configure both nodes to synchronize with the timeserver frequently to avoid losing synchronization, which can be done by using the proper clock synchronization protocol. For example, Network Time Protocol (NTP) regularly polls a group of time servers and keeps the system clock in synch from moment to moment [47].

4.3.3.4.2 When the access points do not have access to a time server

As previously discussed, the authorization entity has a counter running for each of the access points in the system. When access points do not have access to a time server, restoring synchronization is done by using the value of the counter broadcasted by the token transmitting access point. In the proposed system, every time a location proof is sent from a device that is granted access to the authorization entity, the value of the counter is stored in the `last_admitted_counter` parameter: a parameter that stores the value of the counter received from a device that was previously granted access to resources. Restoring synchronization is done as follows:

1. If an invalid location proof is received from the user's device, the authorization entity compares the value of the received location proof with the value of the location proofs that are generated using previous counters (starting from the current counter decrementing until the `last_admitted_counter`)
2. If a match is found, the authorization entity updates the local counter with the counter embedded in the location proof. This restores the synchronization between the access point that was out of sync and the authorization entity. Then, access is granted.
3. If no match is found, access is denied.

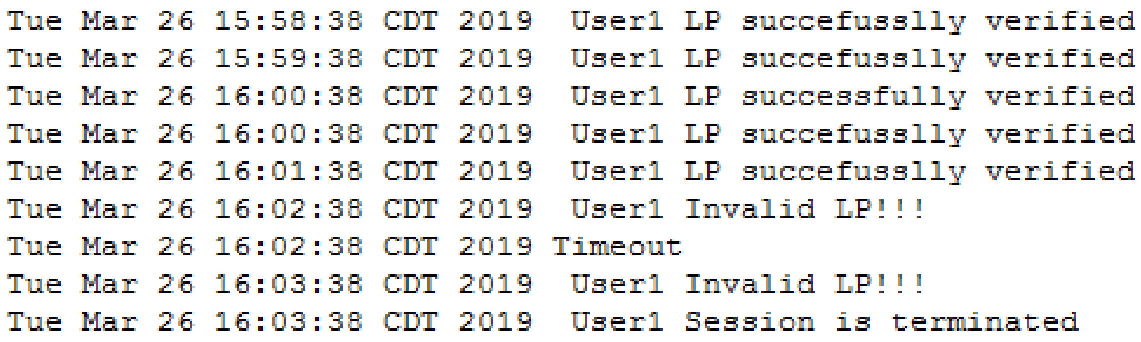
4.3.3.5 *Continuous authentication*

As previously mentioned, the proposed system provides continuous authentication with zero effort from the user, which is done by having the user's device continuously send location proofs to the authorization entity for verification. The authorization entity continuously checks the location proofs received from a user's device and accordingly makes access decisions.

There are two timing parameters in the continuous authentication portion of the authorization entity: a timeout and a check period. The timeout represents the time that the authorization entity waits before verifying the user's credential again if the first authentication attempt fails. The purpose of using the timeout is to overcome any failures at the user's device or the access points by giving them another chance to present valid credentials. The check period parameter is the period that determines how often the authorization entity checks for the latest credentials provided by the user's device and terminate the session if invalid ones are received. The pseudocode below describes how the authorization entity continuously verifies the location proof sent by a user’s device.

```
while true:
        if location_proof_local != location_proof_received:
        sleep timeout
        if location_proof_local!=location_proof_received:
                kill ssh_session(user)
        sleep check_ period
```

As seen from the pseudocode, the authorization entity compares the latest location proof received from the user's device `(location_proof_received)` with the location proof generated locally (`location_proof_local`). If they do not match, the authorization entity waits through the *timeout* and checks again after the timeout period expires. If these values are still not equal after the timeout, the authorization entity terminates the user’s session. In our experiment, SSH is used for connection between the user’s device and the authorization entity. Thus, terminating the user’s session is done by killing the process of the SSH session for that user. The `check_period` specified in the pseudocode is the period in which the authorization entity waits before checking for the location proof again. Figure 4-13 shows the authorization entity system logs for the continuous authentication of a user.



```
Tue Mar 26 15:58:38 CDT 2019  User1 LP succefusslly verified
Tue Mar 26 15:59:38 CDT 2019  User1 LP succefusslly verified
Tue Mar 26 16:00:38 CDT 2019  User1 LP successfully verified
Tue Mar 26 16:00:38 CDT 2019  User1 LP succefusslly verified
Tue Mar 26 16:01:38 CDT 2019  User1 LP succefusslly verified
Tue Mar 26 16:02:38 CDT 2019  User1 Invalid LP!!!
Tue Mar 26 16:02:38 CDT 2019 Timeout
Tue Mar 26 16:03:38 CDT 2019  User1 Invalid LP!!!
Tue Mar 26 16:03:38 CDT 2019  User1 Session is terminated
```

Figure 4-13: The system logs for terminating the session of a user leaving the access area.

As can be seen in the figure, the authorization entity verifies the user’s location proof every 60 seconds. Moreover, when the user did not send a valid location proof, the authorization entity took a 60-second timeout before verifying the user’s location proof again. Then, the user’s session was terminated due to the user failing to send a valid location proof.

4.3.3.6 *Access policy*

The goal of the access policy is to grant different access rights to different users. Access rights are the permissions a user holds to read, write, modify, delete, or access a resource (such as a file). Since SSH is used for authentication in the proposed system, it is possible to configure PAM to take some actions prior to authorization (e.g., mounting a drive). However, access policies can be specified in many ways for various connection methods to resources. Investigating other methods is a potentially interesting question for future research.

4.4 Experiments and Results

In the previous section, we discussed the design specifics. This section presents a number of experiments that were utilized to test the proposed design. In the experiments, the following hardware was utilized:

1. Access points: Netgear R6400 Wi-Fi access points.
2. Authorization entity/resource server: Intel i5 and 8GB of RAM PC server with Ubuntu 18.10 (GNU/Linux 4.18.0-14-generic x86_64) as the operating system.
3. User devices: in most of the experiments, Raspberry Pi with Raspbian 4.14.50-v7+ were used. However, different models of android smartphones were used in one experiment. Raspberry Pi was used as the user device in most of the experiments, due to the flexibility it offers, enabling much faster deployment, and it allows making changes on the fly, making it a great fit for experimental environments.

Experiment 1 analyzed RSSI measurement consistency, while Experiment 2 studied the differences in RSSI measurements between different models of user devices. Experiment 3 tested the access area's definition and performance, while Experiment 4 tested the system's operation under different broadcast periods. Experiment 5 studied the effect of implementing various timeouts, and Experiment 6 tested the effect of token transmission on access point operation (i.e., resource usage). Finally, Experiment 7 tested the system's performance at the edges of access areas, and Experiment 8 studied the system's response when a user leaves an access area.

4.4.1 Experiment1: RSSI Measurement Consistency

In the proposed system, an access area is defined by the coverage of relevant access points. These access points combine to produce an RSSI footprint to define access areas. In our experiments, a user device was placed in a fixed position inside the access area. This device recorded 15,000 RSSI values from an access point in range, with one reading recorded every 10 seconds. The results show that the recorded values are consistent with few anomalies. These anomalies happen for different reasons; for example, power source inconsistencies, signal obstruction, errors in readings, etc., which can cause instant inconsistent readings. Figure 4-13 shows the distribution of the 15,000 recorded RSSIs. As can be seen in the figure, the distribution of the recorded RSSI values is consistent for most of the data.

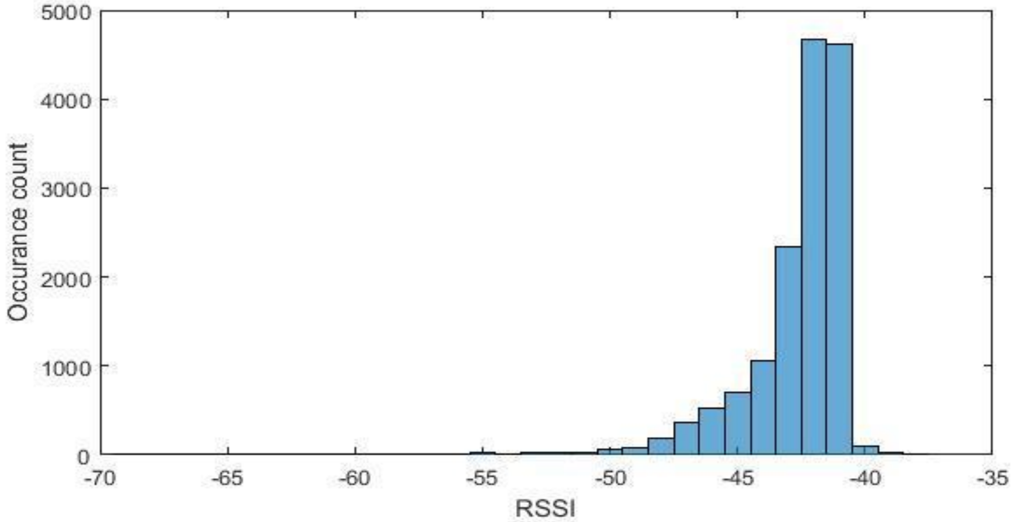
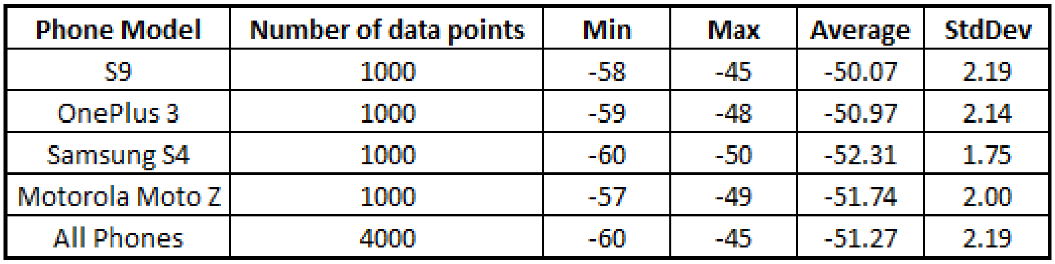


Figure 4-14: The distribution of the RSSIs of an access point recorded by a device.

4.4.2 Experiment 2: Different Models Between Phones

In this experiment, five different models of smartphones were placed next to each other. RSSI data was collected for an access point in range by the five smartphones. The results are shown in Table 4-1.

Table 4-1: RSSI measurements for an access point using different models of smartphones.



Phone Model	Number of data points	Min	Max	Average	StdDev
S9	1000	-58	-45	-50.07	2.19
OnePlus 3	1000	-59	-48	-50.97	2.14
Samsung S4	1000	-60	-50	-52.31	1.75
Motorola Moto Z	1000	-57	-49	-51.74	2.00
All Phones	4000	-60	-45	-51.27	2.19

As can be observed from the results, the RSSI measurements are consistent across the different device models. More details about this are discussed in the next chapter.

4.4.3 Experiment 3: Access Area Definition and Performance

In this experiment, as shown in Figure 4-14, two devices were placed in neighboring offices with three access points in range. These devices recorded all tokens broadcasted by the access points along with their RSSI values. This was done twice: first, with the access points operating at the 2.4 GHz frequency; second, with the access points operating at the 5 GHz frequency. A total of 20,000 RSSI values were recorded for each frequency. The distribution of the recorded data in this experiment is presented in Tables 4-2 and 4-3.

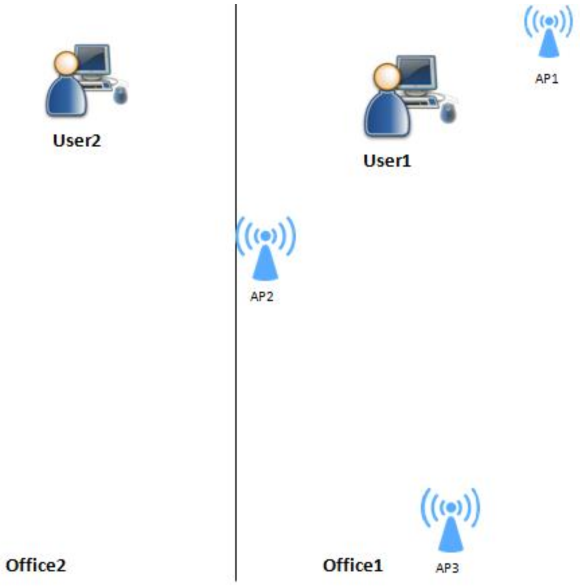
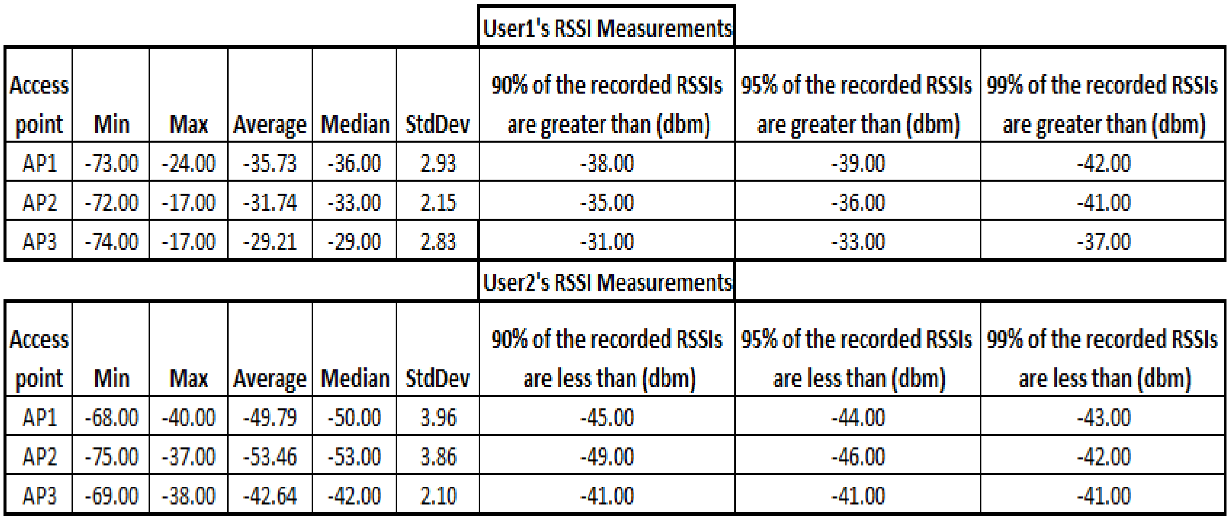


Figure 4-15: Experiment 3 setup.

Table 4-2: The distribution of the recorded RSSIs of the tokens broadcasted by the access points (at 2.4 GHz).



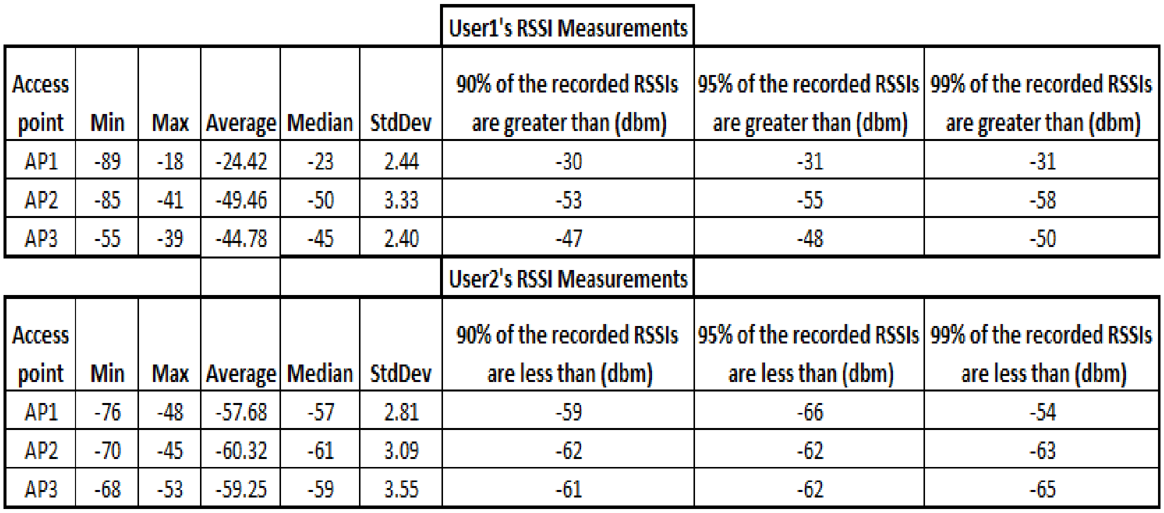
User1's RSSI Measurements

Access point	Min	Max	Average	Median	StdDev	90% of the recorded RSSIs are greater than (dbm)	95% of the recorded RSSIs are greater than (dbm)	99% of the recorded RSSIs are greater than (dbm)
AP1	-73.00	-24.00	-35.73	-36.00	2.93	-38.00	-39.00	-42.00
AP2	-72.00	-17.00	-31.74	-33.00	2.15	-35.00	-36.00	-41.00
AP3	-74.00	-17.00	-29.21	-29.00	2.83	-31.00	-33.00	-37.00

User2's RSSI Measurements

Access point	Min	Max	Average	Median	StdDev	90% of the recorded RSSIs are less than (dbm)	95% of the recorded RSSIs are less than (dbm)	99% of the recorded RSSIs are less than (dbm)
AP1	-68.00	-40.00	-49.79	-50.00	3.96	-45.00	-44.00	-43.00
AP2	-75.00	-37.00	-53.46	-53.00	3.86	-49.00	-46.00	-42.00
AP3	-69.00	-38.00	-42.64	-42.00	2.10	-41.00	-41.00	-41.00

Table 4-3: The distribution of the recorded RSSIs of the tokens broadcasted by the access points (at 5 GHz).



						User1's RSSI Measurements		
Access point	Min	Max	Average	Median	StdDev	90% of the recorded RSSIs are greater than (dbm)	95% of the recorded RSSIs are greater than (dbm)	99% of the recorded RSSIs are greater than (dbm)
AP1	-89	-18	-24.42	-23	2.44	-30	-31	-31
AP2	-85	-41	-49.46	-50	3.33	-53	-55	-58
AP3	-55	-39	-44.78	-45	2.40	-47	-48	-50
						User2's RSSI Measurements		
Access point	Min	Max	Average	Median	StdDev	90% of the recorded RSSIs are less than (dbm)	95% of the recorded RSSIs are less than (dbm)	99% of the recorded RSSIs are less than (dbm)
AP1	-76	-48	-57.68	-57	2.81	-59	-66	-54
AP2	-70	-45	-60.32	-61	3.09	-62	-62	-63
AP3	-68	-53	-59.25	-59	3.55	-61	-62	-65

As seen from Table 4-2, the data shows that setting the accepted range of RSSI values in access area Office 1 for AP1 to -24 to -42 dBm, for AP2 to -17 to -41 dBm, and for AP3 to -17 to -37 dBm results in more than 99% successful authentication for User 1 claiming to be located in Office 1 (the intended access area for User 1). However, these ranges result in more than 99% authentication failures for User 2 claiming to be located in Office 1. These percentages are calculated by analyzing the data received from the user's device at the authorization entity. This can be visualized in Figure 4-15, which shows the probability of each RSSI value of access point AP3 to occur at User1 and User2 locations. As can be seen from the figure, the RSSI measurements for each user fall at a different range for most of the measurements.

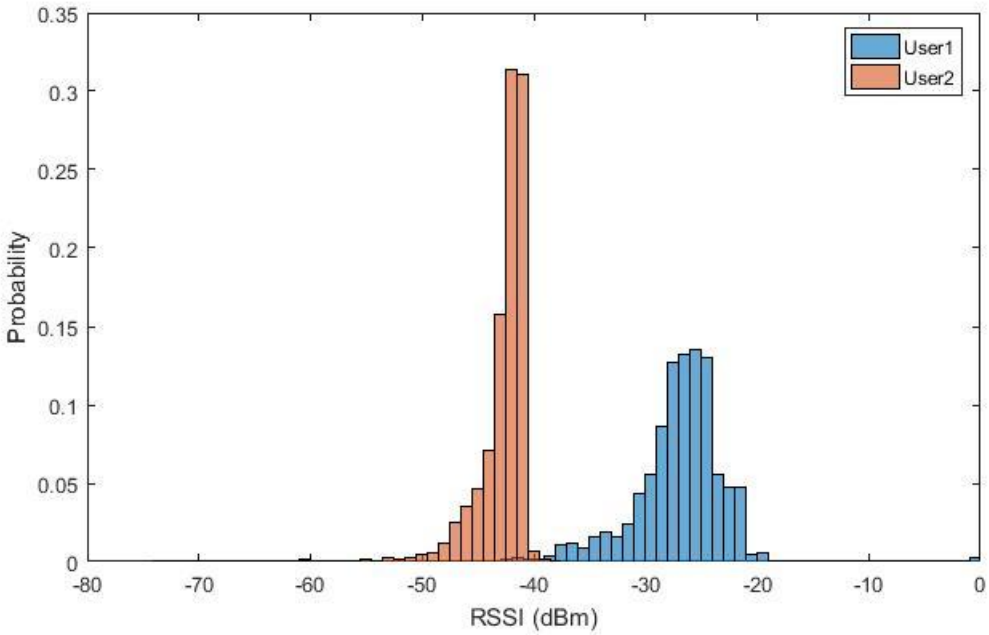


Figure 4-16: The probability of RSSI measurements of access point 3 occurrences at User1 and User2 locations.

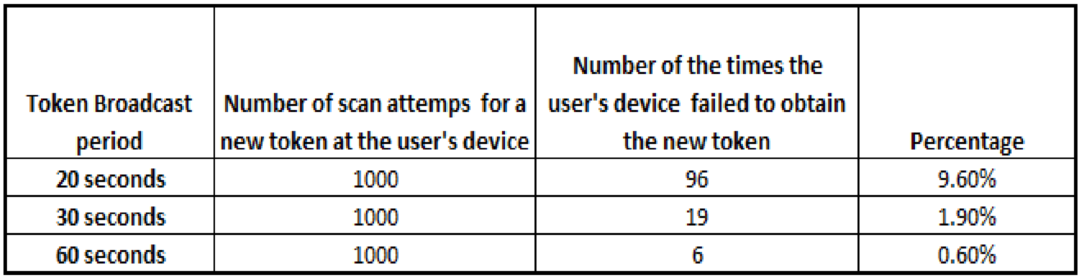
Similarly, for the data shown in Table 4-3, setting the accepted range of RSSI values in Office 1 for AP1 to -18 to -31 dBm, for AP2 to -41 to -58 dBm, and for AP3 to -39 to -50 dBm results in more than 99% successful authentication for User 1 claiming to be located in Office 1 (the intended access area for User 1). However, these ranges result in more than 99% authentication failures for User 2 claiming to be located in Office 1. These percentages are calculated by analyzing the data received from the user's device at the authorization entity

As can be observed from the recorded data, the 2.4 GHz access points seem to have greater RSSI values when compared to the 5 GHz access points. As discussed in chapter 1, this is because the traveling signal path loss is greater for higher frequencies. Accordingly, lower frequency wireless signals have more range, while higher frequency signals do not penetrate solid objects like walls and floors as well as lower frequency signals. Thus, using the 5 GHz band for token transmission in the proposed system will give better results in terms of limiting access to resources to specific access areas.

4.4.4 Experiment 4: Modulating Broadcast Period

The goal of this experiment is to find an acceptable working broadcast period. In this experiment, different broadcast periods were tested: every 20 seconds, every 30 seconds, and every minute. The user's device was then configured to collect tokens broadcasted using these periods. As can be seen in Table 4-4, when a new token was broadcasted every 20 seconds, the user’s device failed to obtain the new tokens 9.6% of the time. Moreover, when a new token was broadcasted every 30 seconds, the user's device failed to obtain a new token 1.9% of the time. However, when the period was s set to one minute, the user's device failed to obtain a new token only 0.6% of the time.

Table 4-4: System’s performance for different broadcast periods.

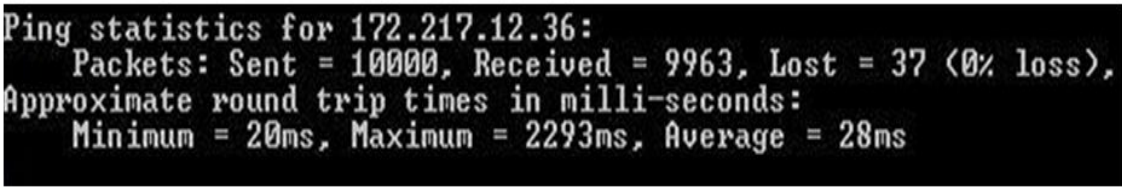


Token Broadcast period	Number of scan attemps for a new token at the user's device	Number of the times the user's device failed to obtain the new token	Percentage
20 seconds	1000	96	9.60%
30 seconds	1000	19	1.90%
60 seconds	1000	6	0.60%

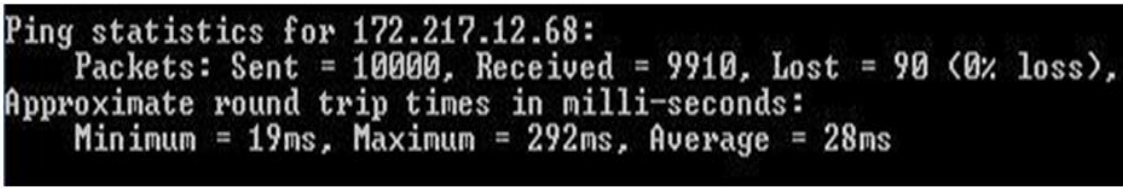
As can be seen from the results, the higher the token generation frequency, the greater the chance that the user's device will fail to obtain a new token. This occurs due to a number of reasons; for example, scan failures at the network interface of the user's device (these sometimes occur) or, more likely, when the user's device scans for broadcasts during the time that an access point's network interface is restarted to apply the new token. In general, increasing the token generation frequency increases the chances for these errors to occur. In real settings, a balance between incorrect RSSI readings and how long a user can exit an access area while still being considered when it must be made. For our experiments, we used 60 seconds as the broadcast period, which we consider to be reasonable in a real-world implementation. This would mean that, in the worst case, a user could maintain access for up to 60 seconds after exiting an access area.

4.4.5 Experiment 5: Access Point Operation

In this experiment, access to production networks was tested while access points were broadcasting tokens in order to evaluate the impact of token generation on the access points. Specifically, we instructed the access points to send a ping (a tool to test the reachability of a host on a network) to test connection continuity and the potential impact of restarting the wireless network interface to apply a new SSID representing a new token. An IP address on the same network as the access points was accessed 10,000 times while the access points were simultaneously transmitting tokens. Moreover, the same experiment occurred with token generation and transmission turned off (to form a comparison). The results are shown in Figure 4-16.



```
Ping statistics for 172.217.12.36:
    Packets: Sent = 10000, Received = 9963, Lost = 37 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 20ms, Maximum = 2293ms, Average = 28ms
```

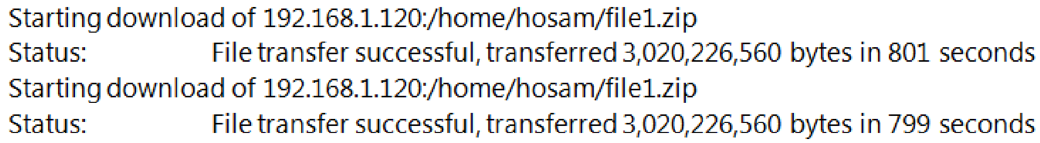


```
Ping statistics for 172.217.12.68:
    Packets: Sent = 10000, Received = 9910, Lost = 90 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 19ms, Maximum = 292ms, Average = 28ms
```

Figure 4-17: Ping statistics, when the token transmitting function was off (top) and on (bottom).

As observed, there is slightly more packet loss (which occurs when a transmitted packet fails to arrive at their destination) when the token transmitting function is on. For example, packet loss was 37/10,000 (0.37%) with token generation turned off while it was 90/10,000 (0.90%) when the token generation was turned on. This happens because the interface is restarted so that the new settings (i.e., change the SSID field of the beacon frame to reflect the newly generated token) take effect every time a new token is generated.

Another experiment was also conducted to test the effect of token transmission on access point operations. In this experiment, a large file was downloaded from a local server using the file transfer protocol (FTP), which is a standard network protocol used for the transfer of computer files between a client and server on a computer network. The large file was downloaded while the token generation was functional. We also tested this with the token generation turned off. As can be seen from the results shown in Figure 4-17, the download speed of the large file was two seconds faster when the token generation was turned off with a similar transmission rate of 3.75 MB/s. However, as can be seen from the experiments, there is no major impact on network performance when the token transmitting function is on, as running the token generation function on the router only caused 0.53% additional packet loss and 0.25% slowdown in transmission speed.



```
Starting download of 192.168.1.120:/home/hosam/file1.zip
Status:          File transfer successful, transferred 3,020,226,560 bytes in 801 seconds
Starting download of 192.168.1.120:/home/hosam/file1.zip
Status:          File transfer successful, transferred 3,020,226,560 bytes in 799 seconds
```

Figure 4-18: FTP download time with and without token transmitting function.

4.4.6 Experiment 6: Timeouts

The device was configured to collect the tokens and the RSSI values of three access points. These access points were configured to transmit a new token every 60 seconds, and consequently, data points were collected every 60 seconds. A total of 10,000 data points were collected. During this time, the user's device could not obtain a token 40 times (0.4% of the time). Ultimately, this is a small percentage. When a timeout is implemented at the authorization entity for when a user device fails to send the latest token, these failures only cause two session terminations. Furthermore, when two consecutive timeouts are implemented, these failures do not cause any session terminations. As seen in Experiment 4, it is highly unlikely for the user’s device to fail in generating and sending a location proof if it is still inside the access area.

4.4.7 Experiment7: Edges

In this experiment, as illustrated in Figure 4-18, a user's device was placed at the edge of an access area inside an office; 2,000 data points were subsequently collected.

The user's device was then placed immediately outside the office (behind a wall – to imitate an attacker), and another 2,000 data points were collected. The goal of this experiment was to test the ability of attackers standing at the edge of an access area to attempt to obtain access to resources by claiming to be inside an access area (when they are not). Table 4-5 shows the distribution of RSSI readings for the user and the attacker.

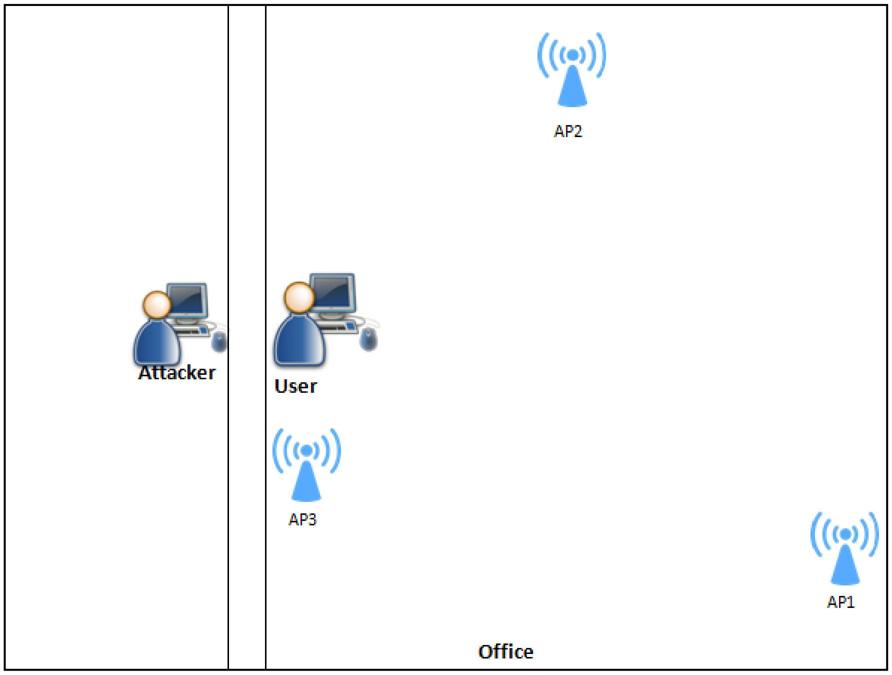
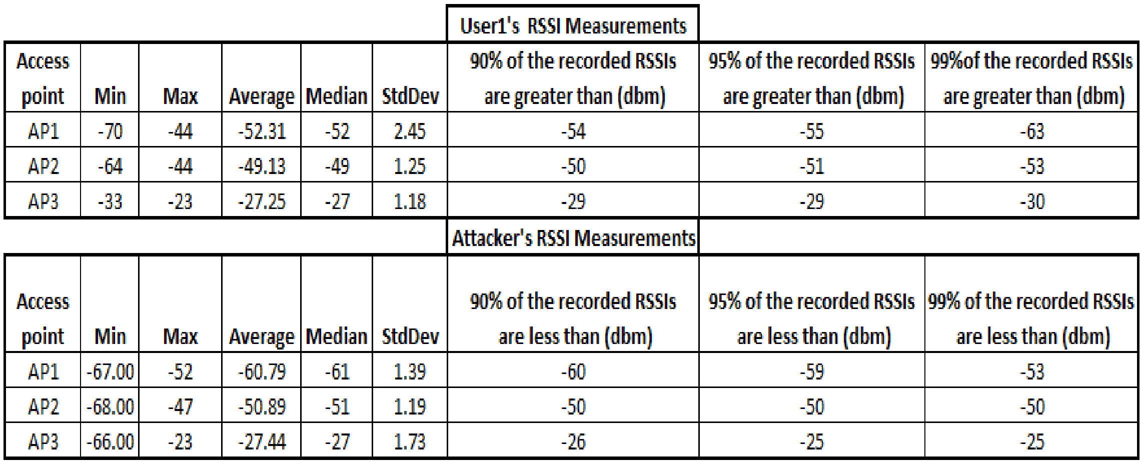


Figure 4-19: Experiment 7 setup.

Table 4-5: The distribution of RSSI readings for the user and the attacker.



						User1's RSSI Measurements		
Access point	Min	Max	Average	Median	StdDev	90% of the recorded RSSIs are greater than (dbm)	95% of the recorded RSSIs are greater than (dbm)	99%of the recorded RSSIs are greater than (dbm)
AP1	-70	-44	-52.31	-52	2.45	-54	-55	-63
AP2	-64	-44	-49.13	-49	1.25	-50	-51	-53
AP3	-33	-23	-27.25	-27	1.18	-29	-29	-30

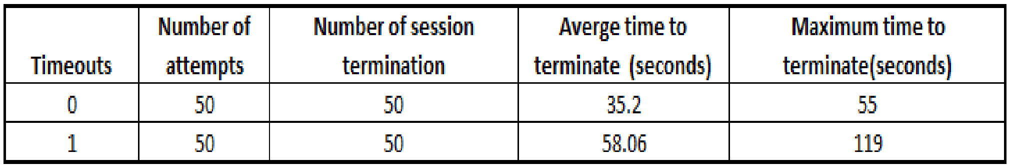
						Attacker's RSSI Measurements		
Access point	Min	Max	Average	Median	StdDev	90% of the recorded RSSIs are less than (dbm)	95% of the recorded RSSIs are less than (dbm)	99% of the recorded RSSIs are less than (dbm)
AP1	-67.00	-52	-60.79	-61	1.39	-60	-59	-53
AP2	-68.00	-47	-50.89	-51	1.19	-50	-50	-50
AP3	-66.00	-23	-27.44	-27	1.73	-26	-25	-25

As can be seen in the results, it is sufficient if RSSI ranges to obtain access inside the office are configured from -55 dBm to -44 dBm for AP1, from -51 dBm to -44 dBm for AP2, and from -23 dBm to -29 dBm for AP3.These values were set by measuring the RSSIs immediately outside the access area and setting RSSI values to thwart attackers. Thus, an attacker claiming to be inside the access area will fail to get access to resources at least 95% of the times. Furthermore, limiting the number of permitted connection attempts will decrease the attacker chances to gain access even more, as it prevents connection attempts from a user after many authentication failures.

4.4.8 Experiment 8: Users Leaving the Access Area

The goal of this experiment is to test if user sessions are terminated once an access area is exited. In this experiment, the token generation period was set to 60 seconds. Furthermore, the user exited the access area 100 times after the first successful authenticating in the access area. In the experiment, 50 trials were done without activating the timeout function discussed in section 4.3.3.5, while the remaining 50 trials were performed with a 30-second timeout. As can be seen in Table 4-6, the system achieved a 100% success rate in terminating the user's session once the access area was exited. Timeout column represents the number of timeouts implemented. Average time to terminate represents the average time it took the authorization entity to terminate the user’s session after leaving the access area for all the attempts. The last column represents the maximum time it took the authorization entity to terminate the user’s session after leaving the access area.

Table 4-6: Authorization entity response to users leaving an access area.



Timeouts	Number of attempts	Number of session termination	Averge time to terminate (seconds)	Maximum time to terminate(seconds)
0	50	50	35.2	55
1	50	50	58.06	119

4.5 Example Scenario

In this section, we discuss an example scenario. In this scenario, three token transmitting access points are used in an access area. As seen in Figure 4-19, three token transmitting access points are used. As previously discussed, the user’s device collects the tokens and uses them to derive a location proof that is later used to authenticate with the authorization entity. In this section, we discuss what happens at each of the nodes. Note that we assume that the access points have access to a time server. thus, a timestep is used in the token generation.

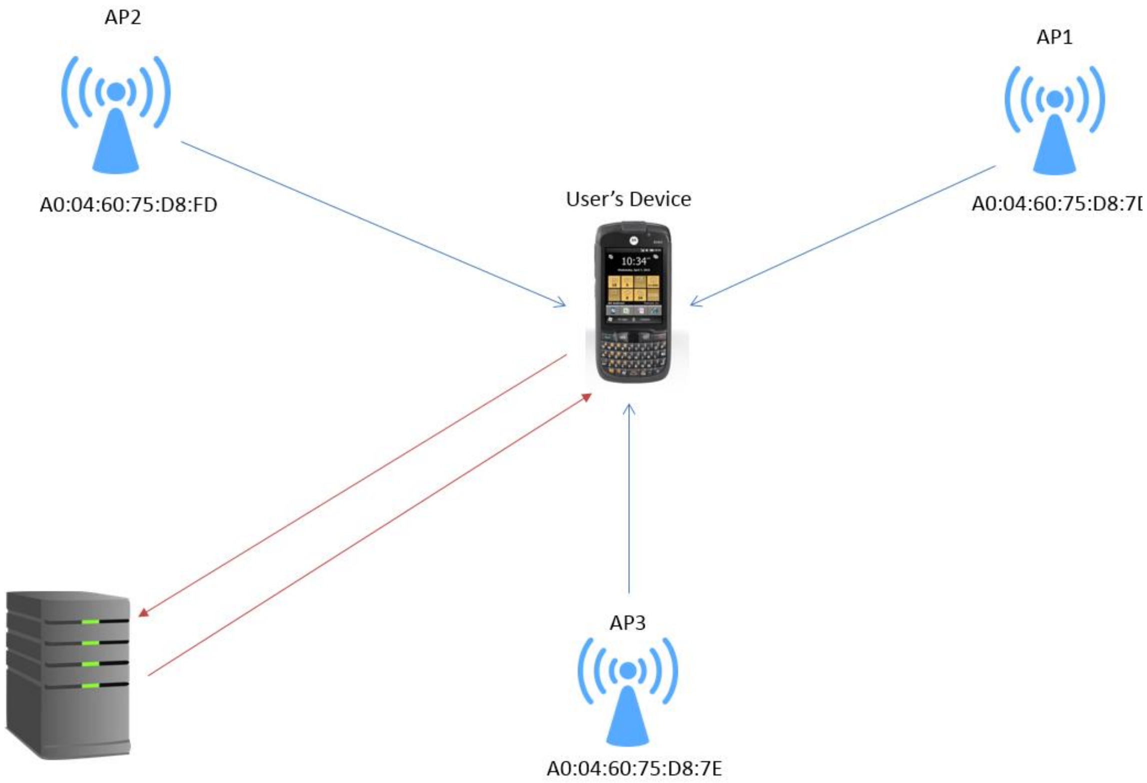


Figure 4-20: The scenario setup.

At the access points:

As previously discussed, each access point generates tokens using a pre-shared secret and a timestamp by using Equation 4-1. For this example, the pre-shared secrets at AP1, AP2 and AP3 are “secret1”, “secret2”, and “secret3”, respectively. In this example scenario, the access points generate a new token every 60 seconds. As discussed in Section 4.3.1.1.1, each access point generates, translates, encodes, and finally transmits the token as follows:

1. When access point AP1 executes, the HMAC function shown in Equation 4-1 using the pre-shared secret mentioned above and the timestamp “201903181043”, the output code

results in the following:

f8651e2f6b5f9dfc3988a4d1b14e7ae0f4b0c6c7cb1014b8c37886871ae2a53c.

2. The result obtained from step one is divided by $1{,}000{,}000$, and the modulo of this division is calculated, which, in this case is $442{,}620$. This is ultimately the token, which the user's device uses in deriving a location proof.
3. Next, this token is translated using addition and multiplication using the constants (296,772 and 215,415) as follows:

$$442{,}620 * 296{,}772 \ + 215{,}415 = \ 131{,}357{,}438{,}055$$

4. The translated token is converted to base 64 which yields MTMxMzU3NDM4MDU1Cg==.
5. Now, the access points add the header that allows the client software to recognize that this broadcast is actually a token, which in this example is "AP_".
6. Finally, the access point modifies the SSID field to MTMxMzU3NDM4MDU1Cg== and broadcasts the beacon frame. Similarly, AP2 and AP3 broadcast MTY4MzM4MTk2OTc1Cg== and MTkwNDcxNzQ5NTA3Cg== simultaneously. As can be seen from the access point broadcasts, despite using the same timestamp, the base 64 values are different due to the unique pre-shared secrets used at each of the access points.

At the user device:

As previously mentioned, the client software is installed on the user's device to handle location proof generation and connecting to the authorization entity. The client software generates the location proof as discussed in Section 4.3.2 and as follows:

1. The client software searches the Wi-Fi scan results for access points with the SSID starting with prefix “AP_” and returns the SSID, the MAC address, and the RSSI values of matching access points. In this example, this step returns AP_MjI3NTU3ODYyNDg3Cg==, AP_ Mzg5OTEyNDk2OTg0Cg==, and AP_NjUxMTQ2OTQ1OTc2Cg== along with their RSSIs and MAC addresses.
2. The client software truncates the SSID to get the encoded token translation. Then, the client software decodes the encoded token translation from base 64 to obtain the translated token.
3. As previously discussed, the translation constants are securely stored in the client software. The client software uses these constants to de-translate output to retrieve the original token, as follows:

$$\frac{131{,}357{,}438{,}055 - 215{,}415}{296{,}772} = 442{,}620$$

   Similarly, tokens are deobuscated for AP2 and AP3 to 567,230 and 641,811.
4. The client software generates the location proof using Equation 3-1. In this example, RSSIs for AP1, AP2 and AP3 are -45 dBm -44, dBm, and -40 dBm respectively. The result is as follows:

   A0:04:60:75:D8:7D442620-45A0:04:60:75:D8:FD567230-44A0:04:60:75:D8:7E641811-40
5. Finally, the client software uses the location proof along with a password to authenticate with the authorization entity over a secure channel. In case access is granted, the user device sends a new location proof continuously, every period.

At the authorization entity:

As previously discussed, the authorization entity is responsible for verifying the user’s credentials and granting or denying the user access to resources. When the authorization receives an access request from the user’s device, which includes the username, a password, and a location proof, the authorization entity verifies the password, then verifies the location proof as follows:

1. Using the pre-shared secrets with the access points in the system, the authorization entity locally generates the same tokens generated by the access points in the system every 60 seconds and stores them in a database. When using the timestamp “201903181043”, locally generated tokens of access points AP1, AP2, and AP3 are 442,620, 567,230, and 641,811 respectively.
2. The authorization entity receives a location proof from the user’s device in real time. Moreover, it is aware of the format of the location proof and it divides the location proof based on the pre-known indices. This returns the MAC addresses, the tokens, and the RSSIs of access points as follows:

   A0:04:60:75:D8:7D | 442620 | -45 | A0:04:60:75:D8:FD | 567230 | -44 | A0:04:60:75:D8:7E | 641811| -40.

   $MAC_{AP1}\ |Tk_{AP1}|RSSI_{AP1}\ |MAC_{AP2}\ |Tk_{AP2}|RSSI_{AP2}\ |MAC_{AP3}\ |Tk_{AP3}|RSSI_{AP3}$
3. Using the information returned from Step 2, the authorization entity identifies the access area following the steps discussed in Section 4.3.3.2.
4. The authorization entity compares the token received versus the one generated locally. Moreover, it verifies if the received RSSIs fall in range of the previously configured ranges of the specified access area. If the received location proof is valid, access is granted. Otherwise, access is denied. If access is granted, the authorization entity continues to check the newly sent location proofs and terminates the user’s session if invalid ones are received.

4.6 Conclusion

In this chapter, different elements of the system were discussed, including their functions and design requirements. Subsequently, design specifics of each of the elements were discussed, including software and hardware specifics. Furthermore, this chapter described the steps that the access points and the authorization entity undergo to generate a token based on the pre-shared secret (which was discussed in two scenarios: when the access points have access to a time server and when they do not). Subsequently, a discussion of the method utilized to verify location proofs at the authentication server was discussed, including performing continuous authentications and granting access based on specified access policy. In the next chapter, the proposed system will be analyzed, and strengths and weaknesses will be identified and discussed. Furthermore, the different parameters that are used to optimize the system's performance will be discussed.

CHAPTER 5

DISCUSSION

In the previous chapter, the design details of the proposed system were covered. In this chapter, the different aspects of the proposed system are evaluated based on the experiments performed. Moreover, the different factors and parameters that play a role in the proposed system's performance are analyzed. Section 5.1 analyzes the proposed system's security and its response to different types of attacks. Section 5.2 evaluates the proposed system from the point of view of user convenience. Section 5.3 discusses the steps taken to ensure the robustness of the system. Section 5.4 provides a discussion of access policies. Section 5.5 summarizes the proposed system's deployment estimated costs to better gauge real-world implementation. We conclude in Section 5.6.

5.1 Security

In general, the main goal of employing multiple-factor authentication is to enhance the security of a system. For example, two-factor authentication adds an additional level of security in case the first authentication factor is compromised, making it harder for attackers to access systems with stolen passwords, for example. In our work, the additional authentication factor is intended to associate an authorization to use resources to the physical location of users to ensure that they can only access specific resources when they are physically located in a specific location. In this section, the security features of the proposed system are analyzed, followed by a study of the system's response to different types of anticipated attacks.

5.1.1 Security Features

We will first examine the security features of the proposed system by analyzing how secure the tokens are generated using the HMAC function. Next, access areas are discussed. Finally, the continuous authentication feature in the proposed system is analyzed.

5.1.1.1 *Token security*

In the proposed system, tokens are generated using the HMAC function with a pre-shared secret along with the current time (or a counter). The SHA256 hash function is used, which is considered secure in a cybersecurity context. In fact, it is used by the Linux operating system for secure password hashing [18]. With current computational resources, it is nearly impossible to brute force a SHA256 hash to obtain the pre-shared secret. In general, HMAC is secure [9]. Furthermore, to enforce the token's security against attacks (e.g., a dictionary attack), tokens are altered (or translated) by performing various arithmetic operations before they are broadcasted in the beacon frame by participating access points. Only users with the client software are able to regenerate the original tokens since only the client software has access to the translation constants. Translation constants are stored securely in the client software (by using source code protection tools). Even though if attackers were able to retrieve the translation constant (e.g., by reverse engineering the client software), they still would need physical access to the access area, a valid username, and, if two-factor authentication is used, an additional authentication factor (such as a password).

5.1.1.2 *Access areas*

For deployments with higher security requirements, accepted RSSI ranges can be set more narrowly to eliminate any chance of unauthorized users obtaining access to a protected resource from outside of a specified access area. In the experiments in the previous chapter, access points with omnidirectional antennas were used. An omnidirectional antenna radiates equal radio power in all directions. However, access areas can be tailored more accurately using directional antennas, which are designed to function more effectively in one specific physical direction. Moreover, shielding paint can be used to block the leakage of access point signals through walls, windows, doors, etc. Combining these strategies can result in more accurate delineation of access areas.

5.1.1.3 *Continuous authentication*

In the proposed system, continuous authentication is necessary. Continuous authentication ensures that users maintain access to resources only when they maintain their presence in specified access areas. This is done by having a user's device continuously send the location proof and requiring the authorization entity to continuously verify it. In the previous chapter, we experimented setting the broadcast period to different values (some experimental values resulted in stable system performance). However, the broadcast period depends on the deployment's needs. The goal of setting the token generation frequency interval (to, for example, every one, two, or five minutes) is continuous authentication to ensure that users will lose access to resources once they leave an access area. As seen in Experiment 8 in Chapter 4, the user's sessions were terminated successfully 100% of the time when the user left the access area.

5.1.2 Cyber-attack Vulnerability Assessment

In this section, the proposed system's vulnerability for some of the common cyber-attacks is assessed and discussed. These cyber-attacks were chosen by analyzing the possible threats on our proposed system and its components. Furthermore, the steps taken to minimize threats from these types of attacks are also discussed. Section 5.1.2.1 focuses on eavesdropping, Section 5.1.2.2 focuses on man-in-the-middle attacks, section 5.1.2.3 looks at attacks that attempt to gain unauthorized access to the access points, Section 5.1.2.4 discusses the wormhole attack, and Section 5.1.2.5 focuses on the relay attack. These are described below.

5.1.2.1 *Eavesdropping*

In this type of attack, an attacker eavesdrops on access point broadcasts in order to obtain tokens and use them to attempt to authenticate and gain access to a resource. In the proposed system, RSSI values are verified if they correspond to an access area. Access is denied if they do not. However, if an attacker is inside an access area, then the client software is necessary in order to successfully extract tokens from the broadcasts. Tokens can still be sniffed but attackers cannot make sense of them without the client software. In the case of two-factor authentication, the attacker will need a second authentication factor in order to be able to successfully obtain access to resources.

5.1.2.2 *The man-in-the-middle attack*

The man-in-the-middle attack occurs when an attacker is inserted in the system and secretly intercepts communication occurring within the system. Often, such an attacker transmits (and possibly even alters) communication between two parties who believe that they are directly communicating with each other. The result is an appearance of normal communication. The case where an attacker is intercepting communication between access points and a user's device was discussed in the previous section. However, an attacker may also intercept the communication between the user's device and the authorization entity. In our experiments, the SSH protocol is utilized for communicating between a user's device and the authorization entity. This protocol is intrinsically secure and effectively thwarts man-in-the-middle attacks. In practice, however, SSH may not be the protocol that is utilized between users and the authorization server. Regardless, the protocol used may utilize standard asymmetric cryptography through a key exchange in order to mitigate man-in-the-middle attacks.

5.1.2.3 *Targeting the access points*

In this type of attack, an attacker aims to gain unauthorized access to the access points. The goal is for the attacker to compromise an access point and (potentially) obtain the pre-shared secret that is used to generate the tokens. An access point’s management console is typically protected using strong credentials (e.g., username and password). Nonetheless, it is still vulnerable. This type of attack can be minimized by disabling remote management on the access points. This requires direct physical access to the access points in order to manage them. Furthermore, it is possible to encrypt sensitive files stored on the access point. Hence, even in cases of unauthorized access, an attacker will not be able to obtain anything useful as it relates to potentially compromising our proposed LBAC system. Encryption of sensitive information on an access point can be done by using file encryption tools such as GNU Privacy Guard (GPG – on Linux).

5.1.2.4 *The wormhole attack*

In the proposed system, a wormhole attack can occur when an attacker receives the broadcasts at a point inside the access area. To mount this attack, an attacker forwards broadcasts to a rogue device that is located outside of the access area and attempts to use the tokens to obtain access to resources when located outside of the access area. However, in our proposed system, this attack is only possible if the attacker has the client software installed, and is, therefore, able to obtain the tokens from the broadcasts. In general, wormhole attacks are a tricky problem in wireless networks. Despite the challenges in detecting and preventing wormhole attacks, such attacks are limited as the attacker needs physical access to the access area, needs the client software, and, if using two-factor authentication, an additional factor (such as a password).

5.2 User Convenience

Multi-factor authentication enhances the security of a system; however, it usually comes at the expense of user convenience. In the proposed system, the extra authentication factor does not compromise user convenience as it is automatically managed by the user's device. Therefore, continuous authentication occurs without any burden on the users of the system. This is the most significant strength of the proposed LBAC system. It does not require extra steps on the part of the user because the location proof is automatically extracted and sent to the authorization entity by the user's device. This is done by having the user's device scan in the background for participating access points, generate location proofs from those broadcasts, and send them to the authorization entity (repeatedly, ensuring continuous authentication). As seen from Experiment 5 in the previous chapter, the proposed system does not affect an access point's main function to provide access to the network (if this is desired). That is, existing access points can be used to implement the proposed system without compromising their original intended purpose. During our experiments, the access points were used to obtain access to the Internet while generating and broadcasting tokens, with no observable negative side effects (e.g., poor performance).

5.3 Robustness

It is essential for an access control system to have robust operation since failures may potentially hinder a user's ability to utilize the system. In this section, the steps taken in the proposed system to enhance its operational robustness are discussed. Section 5.3.1 examines RSSI fluctuations and Section 5.3.2 covers synchronization loss. Finally, Section 5.3.3 analyzes the token generation frequency.

5.3.1 RSSI Fluctuations

Through various experiments, we observed that RSSI values are fairly consistent. However, anomalies sometimes appear. Because RSSI values are used in the authentication process, it is important to address these anomalies to prevent them from negatively affecting the system. Thus, a timeout is utilized to improve the system's robustness. In the case that an abnormal RSSI value is recorded, the authorization entity will go in a timeout state. It will then check the RSSI values again before terminating a user's session. Moreover, as discussed in Chapter 4, multiple timeouts could increase the overall robustness of the system. An experiment showed that fewer service interruptions are encountered when using two timeouts than using one because providing more chances for users to present valid credentials decreases the chance of them being disconnected.

After testing multiple devices in Experiment 3 (in Chapter 4), results proved consistent with respect to RSSI values measured across different smartphones. This is due to the fact that the internal omnidirectional antennas of mobile phones are necessarily small. Thus, the gain value of small omnidirectional antennas is theoretically limited [16], and in practice does not exceed 5 dB. Gain values of 2 dB or 3 dB are common. Moreover, the Federal Communications Commission (FCC) has limits on Wi-Fi antennas gains [22]. This combines to effectively assure similar RSSI readings across typical mobile devices used by users in the proposed LBAC system.

5.3.2 Losing Synchronization

In the proposed system, tokens are generated at participating access points and at the authorization entity. The same tokens can only be generated if the access points and authorization entity are synchronized. As previously discussed, this can be achieved by using a timestamp or a counter. However, it is possible that synchronization is affected, for example, through power loss at either the access points or the authorization entity. Restoring synchronization is straightforwardly performed when using a timestamp to generate a token. On the other hand, as discussed in the previous chapter, synchronization when using a counter to generate a token is restored by embedding a synchronization parameter in the location proof, which is then used by the authorization entity to update the local counters so that they match the ones at the access points.

5.3.3 Token Generation Frequency

In the proposed system, new tokens are transmitted by the access points constantly to ensure that a user's device is still located in an access area. When access points transmit tokens, their wireless network interfaces are restarted. This forces a change of the SSID (which contains the generated token) to take effect. The wireless network interface will be down for a short time, which will prevent a user's device from scanning for broadcasts from the access point. In Experiment 4 in the previous chapter, the frequency at which new tokens were generated was set to different values, and it was found that more robustness is achieved when setting frequency to at least one minute. Furthermore, as seen in Experiment 8 in the previous chapter, setting the frequency to one minute achieved a 100% success rate in terminating the users leaving the access area session in less than one minute.

5.3.4 Access Point Operations

The proposed system utilizes existing IEEE 802.11 infrastructure and adds a novel LBAC element; therefore, it is essential that it does not affect the existing functionality of the access points. As discussed earlier, the proposed system can create and use a virtual interface on each access point to transmit tokens. Thus, the physical interface can still be used for network access as usual. Moreover, access points with more than one operating frequency can devote one to the LBAC and others to normal networking operations. As shown in Experiment 5 in Chapter 4, configuring the access point to transmit tokens does not significantly impact the access point's functionality to provide access to the network.

5.4 Cost

Since the proposed system utilizes the existing IEEE 802.11 infrastructure, no new hardware is required. However, the access points used must support firmware upgrades so that advanced features can be enabled. This is typically supported in most mid-priced access points that can be purchased in the market today.

5.5 Location-based Policy

Location-based access policies are mainly managed by the operating system of the server providing resources. Most operating systems allow configuring different access rights to different users. In the proposed system, SSH along with Linux PAM were utilized for the connection between the user's device and the authorization entity. Linux PAM allows integrating various access policy functionalities such as mounting/unmounting the user’s home directory and restricting/enabling the services available to the user. However, more advanced location-based access policies can be implemented using different operating systems, which can be a question for future research.

5.6 Conclusion

In this chapter, different aspects of the proposed system were evaluated. Moreover, different configurations and parameters that play a role in the proposed system's performance were examined, and the system’s response to the most common attacks was assessed. This chapter demonstrated that the proposed LBAC system is adequately secure, convenient, robust, inexpensive, highly configurable, and compatible with implementing access policies. The next chapter will summarize this work and explore possible future directions and research that can further extend it.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

LBSs are essential today. They are widely used for a variety of functions such as geo-tagging user-generated content on social media, locating nearby restaurants or ordering food, and obtaining a ride (e.g., using Uber and Lyft). They are also used by advertisers to target advertisements based on the location of their target market's demographics. This work illustrates an important application of LBS: securely controlling access to resources. Recently, it has become more significant since a large proportion of the population uses location-capable mobile devices. Thus, taking advantage of this capability for security applications has also become essential. People use these technologies every day for different functions (e.g., online banking, access control, etc.). LBAC is a security application of LBSs that manages access to resources by utilizing the physical location of users. This research proposes an LBAC system that attempts to take advantage of the current IEEE 802.11 infrastructure, making it directly applicable to the existing, widely used systems in a cost-effective and unobtrusive way for users of the system.

Currently, there is relatively little research that addresses LBAC in an unobtrusive way for users. Most of the current work results in more involvement of users through additional network connections or authentication steps. Inherently, this is inconvenient and can result in inconsistencies: fallible humans are actively involved.

The majority of the research in LBAC addresses either location fixing or access control separately (i.e., no single research endeavor combines the two in an effective, unobtrusive manner). In order to build an efficient LBAC system, it is important to address both as a whole. Hence, providing a unified method to address location fixing and access control simultaneously and in a user convenient way is essential and important.

This research studied the general LBAC architecture and the necessary characteristics that are required to provide an effective and secure LBAC system. Characteristics such as security, user convenience, robustness, and cost combine to produce an effective LBAC system when assured. The work in this dissertation presented a LBAC system design that satisfies the specified requirements. Analyzing and assessing the proposed system's performance substantiated it to meet the design requirements, and showed that the proposed LBAC system is secure, convenient, robust, inexpensive, highly configurable, and compatible with implementing various access policies.

The proposed work details a LBAC system in which a user's location is continuously verified to provide a guarantee that the user can only access resources when physically present within a specified location. Furthermore, and quite importantly, this is done without any involvement from the users of the system. This work adds authentication factor that builds off of existing factors (e.g., username and password). The proposed system is cost-effective and can be directly implemented to the current

IEEE 802.11 infrastructure by simply modifying the software on the access points in place without necessarily requiring additional hardware.

The proposed system supports location-based access policies, which means that the nature of resources granted to a user depends not only on the user's identity but also on the user's location. This allows limiting access to certain resources only when a user is present within a specific location. The proposed system is highly optimizable and allows different configurations that can be used to fine-tune the system to achieve the best result for a specific deployment. Since most of mobile devices in the market today are already location capable, having a secure, unobtrusive, inexpensive, robust, and optimizable LBAC system encourages the addition of the location element to the authentication process in more systems in the future.

6.2 Future Work

This research proposed a LBAC system that can be immediately deployed utilizing existing infrastructure. Hence, one future project can be a practical deployment of the proposed system. Below are some potential future projects:

1- Exploring the different type of hardware and equipment that is supported by the proposed system. This includes experimenting with user devices such as Android and IOS smartphones and exploring the different types of resource servers such as web server, files servers, and so on.

2- Implementing the proposed system in a corporate office setting, in which different users have different access permissions. Furthermore, implementing location-based access policies, in which access to specific resources is controlled by the user’s location along with the user’s identity. For example, limiting access to sensitive resources when a user is inside a specific area.

3- Implementing the proposed system in academic settings to automate the tracking of student attendance. A possible way to accomplish this is by modifying the Wi-Fi access points within the university infrastructure to broadcast tokens, utilizing the personal smartphones of students as user devices, and embedding the client software in the university's smartphone app. This can automate student attendance tracking.

The research presented in this work has raised some additional interesting research questions. We envision several problems arising from this research that can be pursued as future work. The following potential research topics related to this work could be pursued in the future:

1. The potential for using a smartphone as a key to obtain access to another device by configuring the smartphone to transmit tokens based on a pre-shared secret. This can be done by taking advantage of the IEEE 802.11 interface within the smartphone, and by using the “hotspot” function available in most smartphones. By utilizing the same concept of periodic token broadcast, a user could maintain access to a resource only when the smartphone is proximate.
2. The potential for using cellular network infrastructure as opposed to (or in addition to) IEEE 802.11 to expand the proposed system. In this way, tokens are broadcasted by taking advantage of the broadcast channels in cellular technologies (e.g., GSM, UMTS, LTE, and 5G). This would allow implementing vast access areas that can cover larger areas such as streets and neighborhoods.

3. The possibility of taking advantage of the ability of access points to control transmitted power and use variable power transmission patterns to verify that a user is receiving the signal directly from the access point (and, for example, not through a repeater which could be indicative of a man-in-the-middle attack). This could further address relay and wormhole attacks.

This research presents a methodology to build an LBAC system using the current IEEE 802.11 infrastructure. Through this work, interesting problems have been addressed. However, there is still a lot that could be done in the future.

BIBLIOGRAPHY

[1] Acrylic Wi-Fi home, Acrylic Wi-Fi home scanning. acrylicwifi.com.

[2] Agata, Y., Hong, J. and Ohtsuki, T., 2015. Room-level proximity detection using beacon frame from multiple access points. In Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA). pp. 941-945.

[3] Alrahhal, M., Khemakhem, M. and Jambi, K., 2017. A survey on privacy of location-based services: classification, inference attacks, and challenges. Jatit, (1992-8645).

[4] Avdyushkin, M. and Rahman, M., 2015. Secure Location Validation with Wi-Fi Geo-fencing and NFC. In IEEE Trustcom/BigDataSE/ISPA. pp. 890-896.

[5] Azfar R., 2019. U.S. Bank marries geolocation with fraud prevention for Visa cardholders | Retail Dive. Retaildive.com.

[6] Bahl, P. and Padmanabhan, V., 2000. RADAR: An in-building RF-based user location and tracking system. In Proc. IEEE INFOCOM 2000. pp. 775-784.

[7] Bailey, D., Brainard, J., Rohde, S. and Paar, C., 2009. One-Touch Financial Transaction Authentication. In SIGMAP-SECRYPT-ICEB-WINSYS. pp. 5-12.

[8] Bao, L., 2008. Location Authentication Methods for Wireless Network Access Control. In IEEE International Performance, Computing and Communications Conference. pp. 160-167.

[9] Bellare, M., 2006. New Proofs for NMAC and HMAC: Security Without Collision-Resistance. In Annual International Cryptology Conference. CRYPTO.

[10] Berbecaru, D., 2011. LRAP: A Location-Based Remote Client Authentication Protocol for Mobile Environments. In 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing. pp. 141-145.

[11] Bi, Wang, Cao, Qi, Liu and Xu, 2018. A Method of Radio Map Construction Based on Crowdsourcing and Interpolation for Wi-Fi Positioning System. In International Conference on Indoor Positioning and Indoor Navigation (IPIN).

[12] Bianchi, V., Ciampolini, P. and De Munari, I., 2019. RSSI-Based Indoor Localization and Identification for ZigBee Wireless Sensor Networks in Smart Homes. IEEE Transactions on Instrumentation and Measurement, 68(2), pp.566-575.

[13] Bonsor, K., 2018. How Location Tracking Works. HowStuffWorks.

[14] Canlar, E., Conti, M., Crispo, B. and Di Pietro,, R., 2013. CREPUSCOLO: A collusion resistant privacy preserving location verification system. In International Conference on Risks and Security of Internet and Systems. La Rochelle, pp. 1-9.

[15] Cho, Y. and Boa, L., 2006. Secure Access Control for Location-Based Applications in WLAN Systems. In IEEE International Conference on Mobile Ad Hoc and Sensor Systems. pp. 852-857.

[16] Chu, L., 1948. Physical Limitations of Omni-Directional Antennas. Journal of Applied Physics, 19(12), pp.1163-1175.

[17] DD-WRT, 2019. dd-wrt.com.

[18] Drepper, U., 2007. Unix crypt with SHA-256/512.

[19] Ducklin, P., 2019. Serious Security: How to store your users' passwords safely. Naked Security.com.

[20] Dwyer, K., 2019. From Equifax to Facebook: A year for data breaches. Mcall.com.

[21] el Moutia, A., Makki, K. and Pissinou, N., 2007. TPLS: A Time and Power Based Localization Scheme for Indoor WLAN Using Sensor Networks. In IEEE Conference on Technologies for Homeland Security. pp. 117-122.

[22] FCC Rules and Regulations 2.4 & 5 GHz Bands. 2018. Air802.com.

[23] Gaikwad, B., 2014. APPLAUS: A Privacy Preserving Location proof for Location-based service with K-anonymity. International Journal of Computer & Organization Trends, 7(1), pp.29-30.

[24] Gao, C., Yu, Z., Wei, Y., Russell, S. and Guan, Y., 2009. A Statistical Indoor Localization Method for Supporting Location-based Access Control. Mobile Networks and Applications, 14(2), pp.253-263.

[25] Gast, M., 2011. 802.11 Wireless Networks 2nd ed., Sebastopol: O'Reilly Media, Inc.

[26] Grumaz, L., 2015. Location-Based Security for Resource Management. Journal of Mobile, Embedded and Distributed Systems, 3(2067 – 4074).

[27] Guelzim, T. and Obaidat, M., 2008. Novel Neurocomputing-based Scheme to Authenticate WLAN Users Employing Distance Proximity Threshold. In Conference: SECRYPT 2008, Proceedings of the International Conference on Security and Cryptography.

[28] He, W., Liu, X. and Ren, M., 2011. Location Cheating: A Security Challenge to Location-Based Social Network Services. In 2011 31st International Conference on Distributed Computing Systems.

[29] He, X. and Pandharipande, A., 2015. Location-Based Illumination Control Access in Wireless Lighting Systems. IEEE Sensors Journal, 15(10), pp.5954-5961.

[30] Honkavirta V., Perala T., Ali-Loytty S., and Piché R, 2009. A comparative survey of WLAN location fingerprinting methods", Proc. IEEE WPNC, pp. 243-251.

[31] Hsu, Chen, Jaung and Wu, 2018. An Adaptive Wi-Fi Indoor Localization Scheme using Deep Learning. In IEEE Asia-Pacific Conference on Antennas and Propagation (APCAP).

[32] Huseynov, E. and Seigneur, J., 2015. WiFiOTP: Pervasive two-factor authentication using Wi-Fi SSID broadcasts. In ITU Kaleidoscope: Trust in the Information Society (K-2015). pp. 1-8.

[33] Jansen, W. and Korolev, V., 2009. A Location-Based Mechanism for Mobile Device Security. In WRI World Congress on Computer Science and Information Engineering. pp. 99-104.

[34] Javali, C., Revadigar, G., Pletea, D. and Jha, S., 2016. Location fingerprint evidence and authorisation using WiFi channel characteristics. In IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). pp. 1-3.

[35] Javali, C., Revadigar, G., Rasmussen, K., Hu, W. and Jha, S., 2016. I am Alice, I was in Wonderland: Secure Location Proof Generation and Verification Protocol. In IEEE 41st Conference on Local Computer Networks. pp. 477-485.

[36] Jiazhu, D. and Zhilong, L., 2013. A location authentication scheme based on proximity test of location tags. In International Conference on Information and Network Security. pp. 1-6.

[37] Kachore, V., Lakshmi, J. and S.K., N., 2015. Location Obfuscation for Location Data Privacy. In IEEE World Congress on Services. IEEE.

[38] Kulkarni, D. and Tripathi, A., 2008. Context-Aware Role-Based Access Control (CA-RBAC). In Proceedings of ACM Symposium on Access Control Models and Technologies. ACM, pp. 113-122.

[39] Lai, J. and Wibowo, S., 2012. How Service Convenience Influences Information System Success. International Journal of Future Computer and Communication, pp.217-220.

[40] Laplante, P., 2001. Dictionary of Computer Science, Engineering and Technology, Boca Raton, FL: CRC Press.

[41] Leong, Perumal, Peng and Yaakob, 2018. Enabling Indoor Localization with Internet of Things (IoT). In IEEE 7th Global Conference on Consumer Electronics (GCCE).

[42] Luo, W. and Hengartner, U., 2010. VeriPlace: a privacy-aware location proof architecture. In Conference: 18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems.

[43] Makki, A., Siddig, A., Saad, M. and Bleakley, C., 2015. Survey of WiFi positioning using time-based techniques. Computer Networks, 88, pp.218-233.

[44] Malani G., 2019. Location Based Services Market by Component (Hardware, Software, Services, Consulting Services, Managed Services and System Integration Services), Technology (Assisted GPS (A-GPS), GPS, Enhanced GPS (E-GPS), Enhanced Observed Time Difference (E-OTD), Observed Time Difference (OTD), Cell ID, Wi-Fi), Application (Location-based Advertising, Business Intelligence & Analytics, Social Networking & Entertainment, Mapping & Navigation, Local Search & Information, Disaster Management, and Emergency Support) - Global Opportunity Analysis and Industry Forecast, 2014-2022. Allied Market Research.

[45] Mao, G., Fidan, B. and Anderson, B., 2007. Wireless sensor network localization techniques. Computer Networks, 51(10), pp.2529-2553.

[46] Mordor Intelligence, 2018. Location-based Services Market Size - Segmented by Location (Indoor, Outdoor).

[47] ntp.org: Home of the Network Time Protocol, 2015. ntp.org.

[48] Pandey, S., Anjum, F., Kim, B. and Agrawal, P., 2006. A low-cost robust localization scheme for wlan. In WICON '06 Proceedings of the 2nd annual international workshop on Wireless internet.

[49] Pathak, O., Palaskar, P., Palkar,and Tawari, M., 2014. Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points -A Tri-lateration Approach. In Semantic scholar.

[50] Pew Research Center, 2018. Mobile Fact Sheet.

[51] Ren, Y., Oleshchuk, V. and Li, F., 2009. A spatial role-based authorization framework for sensor network-assisted indoor WLANs. In 2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace &Electronic Systems Technology. pp. 782-787.

[52] Rye, G., Seo, C. and Choi, D., 2017. Location Authentication based on Wireless Access Point Information to Prevent Wormhole Attack in Samsung Pay. Advances in Electrical and Computer Engineering, 17(3), pp.71-76.

[53] Saroiu, S. and Wolman, A., 2009. Enabling new mobile applications with location proofs. In Proceedings of the 10th Workshop on Mobile Computing Systems and Applications. Santa Cruz.

[54] Securenvoy – what is 2 factor authentication?. 2018. Securenvoy.com.

[55] Scannell, A., Varshavsky, A., LaMarca, A. and Lara, E., 2009. Proximity-based authentication of mobile devices. International Journal of Security and Networks, 4(1/2), p.4.

[56] Takamizawa, H. and Kaijiri, K., 2009. A Web Authentication System using Location Information from Mobile TelephonesS. In Proceedings of the 8th IASTED International Conference on Web-based Education.

[57] Takamizawa, H. and Kaijiri, K., 2006. Reliable Authentication Method by Using Cellular Phones in Web Based Training. International Journal of Instructional Technology and Distance Learning.

[58] Talasila, M., Curtmola, R. and Borcea, C., 2010. LINK: Location Verification through Immediate Neighbors Knowledge. In International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services. pp. 210-223.

[59] Technavio, 2016. Global Indoor LBS Market 2016-2020.

[60] Technavio, 2017. Global location-based services (LBS) market.

[61] van Rijswijk-Deij., R., 2010. Simple Location-Based One-time Passwords. In PhD thesis. Radboud University.

[62] Vaščák, J. and Savko, I., 2018. Radio Beacons in Indoor Navigation. In World Symposium on Digital Intelligence for Systems and Machines (DISA).

[63] Vile, D., 2006. User convenience versus system security. Theregister.co.uk.

[64] Wang, P. and Luo, Y., 2017. Research on WiFi Indoor Location Algorithm Based on RSSI Ranging. In 2017 4th International Conference on Information Science and Control Engineering (ICISCE). pp. 1694-1698.

[65] Wang, W., Chen, Y. and Zhang, Q., 2016. Privacy-Preserving Location Authentication in Wi-Fi Networks Using Fine-Grained Physical Layer Signatures. IEEE Transactions on Wireless Communications, 15(2), pp.1218-1225.

[66] Wullems, C., Looi, M. and Clark, A., 2003. Enhancing the security of Internet applications using location: a new model for tamper-resistant. In Proceedings of the Eighth IEEE Symposium on Computers and Communications. pp. 1251-1258.

[67] Xiao, L., Yan, Q., Lou, W., Chen, G. and Hou, Y., 2013. Proximity-Based Security Techniques for Mobile Users in Wireless Networks. IEEE Transactions on Information Forensics and Security, 8(12), pp.2089-2100.

[68] Xin-fang, Z., Ming-wei, F. and Jun-jun, W., 2011. An indoor location-based access control system by RFID. In IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems. pp. 43-47.

[69] Zhang, F., Kondoro, A. and Muftic, S., 2012. Location-Based Authentication and Authorization Using Smart Phones. In IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications.

[70] Zheng, Y., Li, M., Lou, W. and Hou, Y., 2012. SHARP: Private Proximity Test and Secure Handshake with Cheat-Proof Location Tags. In European Symposium on Research in Computer Security. pp. 361-378.