AMERICAN UNIVERSITY OF BEIRUT

ARTIFICIAL INTELLIGENCE FOR SECURITY OF IOT DEVICES

by
HADI SAMIR ABDUL GHANI

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
July 2022

AMERICAN UNIVERSITY OF BEIRUT

ARTIFICIAL INTELLIGENCE FOR SECURITY OF IOT DEVICES

by

HADI SAMIR ABDUL GHANI

Approved by:

Type text here
Ali Chehab

Prof. Ali Chehab, Professor Advisor Type text here
Electrical and Computer Engineering

Ali Hussein (Jul 26, 2022 09:28 GMT+3)

Dr. Ali l Hussein, Doctor Co-Advisor
Electrical and Computer Engineering

fadi zaraket (Jul 26, 2022 12:57 GMT+3)

Prof. Fadi Zaraket, Professor Member of Committee
Electrical and Computer Engineering

Rouwaida Kanj
Rouwaida Kanj (Jul 26, 2022 20:43 GMT+3)

Prof. Rouwaida Kanj, Professor Member of Committee
Electrical and Computer Engineering

Date of thesis defense: July 21, 2022

AMERICAN UNIVERSITY OF BEIRUT

THESIS RELEASE FORM

Student Name: ___Abdul Ghani__________Hadi________________Samir____________
Last First Middle

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of my thesis; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes:

☒ As of the date of submission

□ One year from the date of submission of my thesis.

□ Two years from the date of submission of my thesis.

□ Three years from the date of submission of my thesis.

Hadi Abdul Ghani

_________________27 July 2022________

Signature Date

ABSTRACT
OF THE THESIS OF

Hadi Abdul Ghani for Master of Engineering
Major: Software, Networking, Security

Title: Artificial Intelligence for Security of IoT Devices

Physical Layer Security relies on detecting suspected behaviors from the communicated device while authenticating it. It makes use of physical layer attributes to secure the communication. Examples of these attributes are the Received Signal Strength (RSS), and Channel State Information (CSI). This work aims to achieve secure communication between Internet of Things (IoT) devices using Artificial Intelligence while relying on some attributes of the physical layer. In particular, the model will be based on time-series measurements from previous and shared states of RSS, for example. Previous work in this domain included the development of a model to predict the location from such measurements. In this work, we will consider these measurements, specifically, time series measurements, for fingerprint authentication. We will develop a model that captures the fingerprint of the transmitter device and authenticate it based on previous measurements and using different types of machine learning algorithms, which will be trained first to compare fingerprints and the training will continue when receiving authenticated fingerprints. Our contribution is mainly in the authentication process where we implement a model on the receiver side that authenticates the communication by authenticating both legitimate parties using a secret key, while accounting for the environmental effects and movements on the communication link.

TABLE OF CONTENTS

ILLUSTRATIONS

Figure

CHAPTER I

INTRODUCTION AND THESIS OBJECTIVE

Nowadays, the number of IoT devices is increasing exponentially and it is expected, as shown in Fig.1, to reach 25.44 billion devices by 2030. With such tremendous increase in interconnected IoT devices, it becomes critical to secure these connections from any unauthorized users to steal its critical information. As Machine Learning becomes a hot topic in the market and its successful implementation in different topics like medical diagnosis, predicting market prices and many other topics, it appears that it can be used in the physical layer security to ensure the privacy of users while communicating.

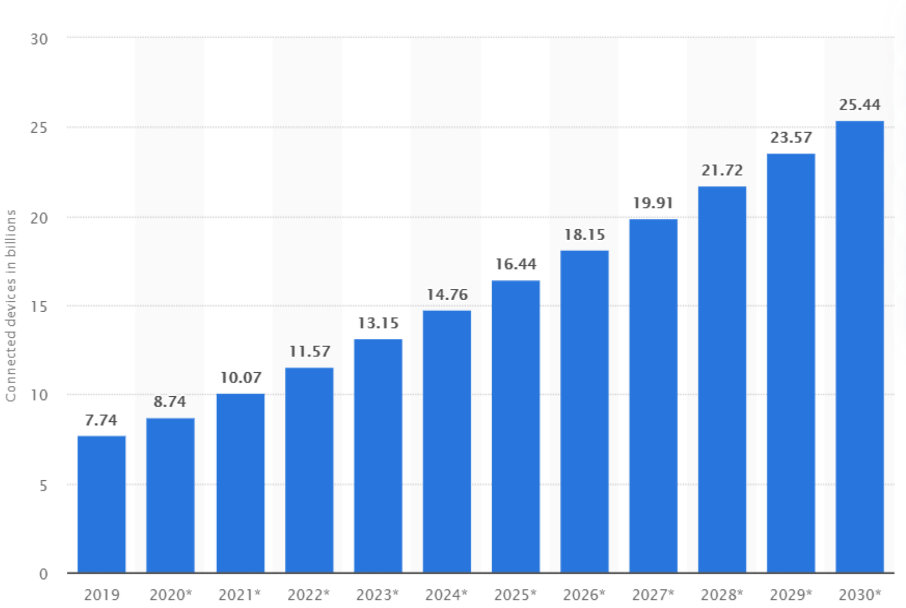


Figure 1 IoT active devices connections between 2019 and 2030

The authentication processes rely on cryptographic techniques using digital secret keys [1], which have many weak points especially with the rapid growth of low-cost devices that makes it easier to crack the security key from the intercepted signal of standardized and static security protocols. In addition, management of security keys requires high computational power to generate, distribute, extract the messages using it, and to revoke the keys, which sometimes causes delays while communicating and executing other tasks. This delay may cause the failure of different communication lines caused by the high probability of disagreement between devices when it increases [2] and it causes critical problems with delay-sensitive devices like vehicles, medical tools that may affect people's life.

The aim of this thesis is to develop a model and train it to monitor the different attributes of physical layer security (PLS) such as the Received Signal Strength and the Channel State Information to either authenticate the transmitter or reject the connection. Specifically, we aim to compare the time series measurements of the PLS attributes with the previous authenticated one and check for minimal changes to authenticate a device. Common work in this field involves applications such as predicting the longitude and latitude of the access point from the RSS using Convolutional Neural Networks (CNN) [3], the proposed idea provides 100% accuracy for building and floor prediction with a mean error in coordinates 2.77 m, [4] which achieves good results; it is based on different measurements such as RSS, CSI, Angle of Arrival (AoA), Time of Flight (ToF) and Return Time of Flight (RToF) [5] where the authors used CSI to improve the accuracy of AoA problem caused by the Signal-to-Noise Ratio (SNR), all of whom focuses on the RSS to predict the location and in certain papers it adds more measurement to improve its accuracy. In the previous works, the goal was to predict the location from different attributes by studying data collected offline. Such an approach does not authenticate the transmitters.

Furthermore, previous works aimed to predict the location of the transmitter using RSS measurements with CNN [3], which targets only the position. In [6], it was proposed to use a novel location signature CSI-MIMO (Channel State Information – Multiple Input Multiple Output) and to use the magnitude and the phase of CSI. The results showed an improvement on the accuracy from using the RSS only. As [5] and [6] showed that CSI improves the accuracy of predicting the location, and since the mean error is in the range of 2 to 5 meters, we will assume that the predictor will be at any point in a circle area of the predicted location of the model with a radius of 5 meters.

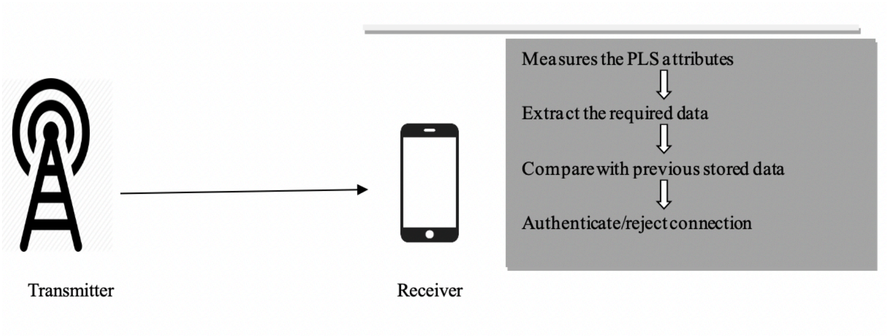

Measures the PLS attributes
Extract the required data
Compare with previous stored data
Authenticate/reject connection
Transmitter
Receiver

Figure 2 Illustration of the target problem

As Figure 2 shows, our target problem is the authentication of the messages sent from the transmitter to the receiver.

The outline of the proposal is as follows: section 2 will cover the literature review of different physical layer security attributes, the Random Forest machine learning algorithm, and the related works. Section 3 will cover the methodology of the proposed idea in addition to the approach followed and the challenges. Section 4 will cover the implementation and datasets. The results are presented in section 5 with the comparison of different algorithm and usage of different metrics. Finally, the conclusion is covered in section 6.

CHAPTER II

LITERATURE REVIEW

A. Physical Layer Security

Unlike the cryptographical approaches, physical layer security does not require high computational power nor complex algorithms. Instead, it takes advantage of its different attributes such as noise, interfering signals or fading to boost the signal received at the legitimate receiver and degrade it at the eavesdropper side. Physical layer security has many advantages over other security techniques, starting with the simplicity of the associated algorithms when compared to the encryption-based methods. In addition, PLS does not rely on encrypting/decrypting data, which overcomes the difficult task of distributing and managing secret keys. It can fully exploit the characteristics of the wireless channels, which offers a flexible configuration to implement different security methods.

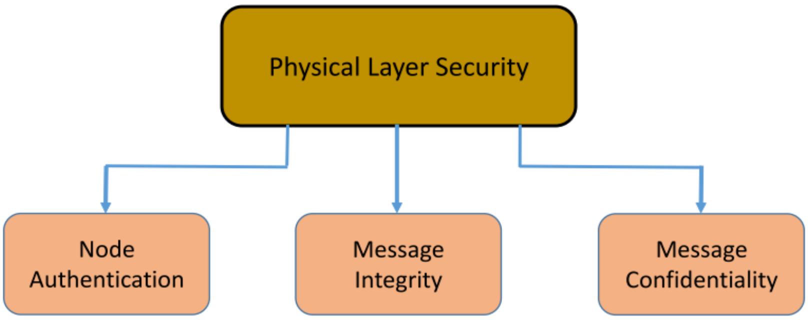


Figure 3 The three main operations of PLS [7]

PLS can provide different operations to make use of all the channel's properties. In [7], the authors consider three main operations provided particularly by PLS:

1. *Node Authentication*: PLS authentication protocols usually consist of two stages: The enrollment stage, which occurs offline; unique characteristics of a user are measured, and a hashed version of these measurements is stored at the verified side along with related helper information. In the release state, new measurements are sent to the verifier where the latter uses the helper to generate the hashes and perform the comparison.
2. *Message Integrity:* A major requirement to pass the PLS is the message integrity, which is considered more important than Message Confidentiality since the message may not be secret, but it should be authenticated to make sure that the eavesdropper did not change the content of the message.
3. *Message Confidentiality:* It relies on two distinct approaches, 1) the wiretrap model approach, which relies on the characteristics of the receiver's channel and 2) the secret key generation approach, which generates a key from randomness by exploiting an area (characteristics of the channel) shared between the legitimate users.

The node authentication (NA) uses Physical Unclonable Functions (PUFs), which rely on unclonable characteristics of the hardware inherited from inevitable variations during the fabrication process. These characteristics are unpredictable and considered as a fingerprint of the device. On the hand, the NA may use Biometrics since due to the noise measurements, the biometric print is updated as discussed above following the 2 stages.

The PLS relies on different attributes while operating as mentioned in [8]; the authentication process relied on different attributes such as RSSI, CSI, Carrier Frequency Offset (CFO), Round-Trip Time (RTT), in-phase-quadrature phase imbalance (IQI), and other attributes:

1. *Received Signal Strength (RSS):* It is the strength of the signal measured at the receiver's antenna. It is determined by the transmission power, distance between the transmitter and the receiver, and the radio environment.
2. *Channel State Information (CSI):* It refers to the channel properties of the communication link that represents how the signal propagates from the transmitter to the receiver at a certain carrier frequency. It is a 3-dimensional matrix with complex values that consists of the amplitude attenuation and phase shift of the multipath Wi-Fi channels.
3. *Round-Trip Time (RTT):* It is the time a single packet takes to be reach the receiver and come back to the transmitter. It can be affected by propagation delay, processing delay, queuing delay, and encoding delay.

B. Artificial Intelligence

1. Random Forest

Random Forest is a classifier consisting of different trees that grow in randomly selected subspaces of the trained data [9]. They are fast and easy to implement, in addition to the high accuracy of predictions, and it can handle many inputs, which fits the requirements of the large number of RSS measurements needed to get the location of the transmitter. Random Forest has consistency in the prediction field as shown in [10] where it gives information about the full conditional distribution of the variables. Finally, in numerical examples it is shown that the random forest algorithm is a competitor in the field of power prediction.

Compared to the other machine learning algorithms used in this field (CNN, deep learning, KNN, etc...), Random Forest has a high computational speed which is an important spec for our model. In addition, the high accuracy achieved by Random Forest in predicting the indoor localization using the RSS values (approximately 95%). It is consistent in the prediction as discussed above and it fits a large number of inputs which is a perfect fit for our model. All in all, Random Forest is one of the most efficient algorithms to be used for the IoT devices in many aspects discussed above and to overcome the different challenges of these devices (power consumption for example).

In [11], a random forest approach was implemented to tackle the indoor localization issue, using CSI where the model is trained offline, and a series of experiments were conducted in an office to extract the results. Compared to K-Nearest-Neighbor the random forest showed a significantly higher classification accuracy and lower mean location error. Thus, the proposed implementation offered high performance in accuracy, robustness and workload.

2. *Logistic Regression*

Since indoor localization became a raising demand in daily activities, the authors in [12] proposed a logistic regression approach under the deep learning framework to tackle the indoor localization issue which was able to achieve an accuracy of 97.2 cm in the laboratory environment. As the traditional models can achieve an accuracy of less than 1 meter by using multiple CSI, but the computational overhead is high. To purpose of this paper was to address this issue using logistic regression instead of the traditional classification technique.

In addition, the authors in [13] analyzed a shopper's behavior using CSI of WiFi instead of using video surveillance due to the high cost and privacy. The classifier was trained by a user moving in different states as show in figure 4 below.

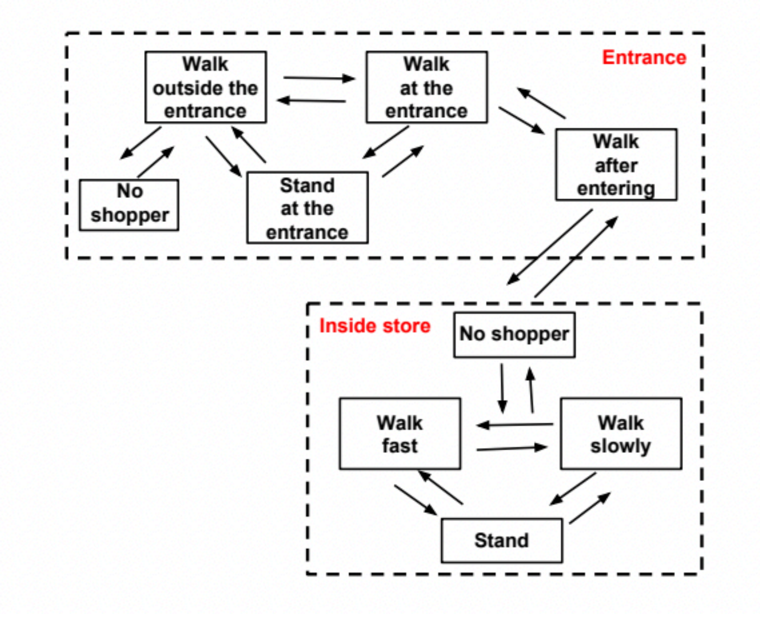


Figure 4 Shopper's states transition diagram [13]

The authors used 2 machine learning algorithms: decision tree and logistic regression. The classification results of 95% for logistic regression to classify the different states of the shopper based on the CSI.

3. *KNN Classifier*

In [14], the authors proposed an indoor localization solution based on CSI using K-nearest-neighbor where a high accurate positioning were achieved. The input features were the amplitudes of the CSI which were processed to reduce the noise.

The model presented achieved a Mean Square Error of 2.4 cm which outperformed 3 different models based on deep learning algorithm.

4. Support Vector Machines

In [15], the authors proposed a human flow recognition system based on CSI and support vector machine. The feature values used are the amplitude and phase extracted from the measured CSI. The proposed implementation achieved high results as an accuracy of 100% for the humans passing in the same workflow and 99% for 2 directions.

As a result, CSI measurements are used in most of the indoors localization applications that resulted in high accuracy in the field. Thus, CSI is a required field that can be used to achieve high accuracy in the related implementation. In addition, based on the authors in [16]'s findings, it is confirmed that CSI and RSS separately are able to provide high accuracy in the field. However, if we fuse both properties (RSS and CSI amplitudes), it will result in even higher accuracy based on their results where they used a KNN model and combined RSS and CSI amplitudes as features for the model.

C. Public Key Technique

Previously, cryptography was based on using symmetric keys to ensure the security of data exchanged between the receiver and the transmitter, but symmetric key was lacking the exchange of the key between the legitimate parties. Thus, researchers invented the public key encryption to overcome this issue [17]. Later, public key encryption became the most widely used cryptography type used.

Public key does not require exchanging the key between the sender and receiver by having a pair of keys, it enables the exchange of key in unsecured network. It is the best solution for the cryptography type that can be used by the sender and receiver. Figure 5 will illustrate the process of encryption and decryption using public key.

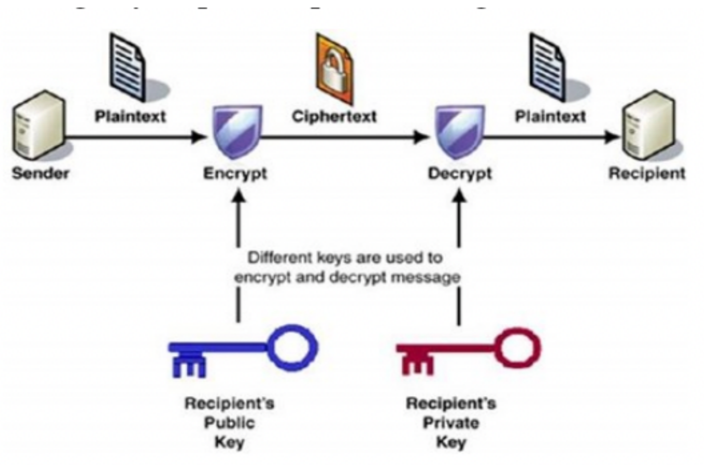


Figure 5 Process of encryption and decryption [18].

The process is as follow:

1. The receiver will create the keys (public and private) to use in the encryption process.
2. The private key will be generated based on the public key, and both are mathematically related.
3. The receiver will send the public key to the transmitter to encrypt the data.
4. The transmitter will encrypt the data and send it to the receiver.
5. The receiver will decrypt the encrypted data using his private key.

However, each type has its own vulnerabilities, and for the public key its vulnerabilities can be summarized based on [19] and our implementation:

1. Randomness in key generation:

To ensure that the system is safe from the attackers, it is required to have a randomness in the prime numbers used in the generation process. The used numbers should be sufficient to generate an unpredictable key. However, a random number generator may work well in some situations, but it is not useful for the other [20].

In our case, the system will be used for a short period of time to gather enough data of the surrounded physical environment for the model to be trained and implemented. Thus, this vulnerability won’t affect our system as it will drop the keys after implementing the system which won’t create any challenge such as creating the unpredictable keys or not working due to randomness in some cases.

2. Man in the Middle Attack:

The source of the data received at the received side cannot be guaranteed that it is from a legitimate source, as if the attacker was able to decrypt the data between the sender and receiver then the attacker will act as a receiver for the legitimate sender and the legitimate sender for the receiver. The process followed by the attacker to have this scenario is by sending his public key to the legitimate sender and receive the public key of the receiver. In this scenario, the authentication failed by the first step. To handle it, we can use a third legitimate party (server) assumed to know both the legitimate sender and receiver.

3. Time Spent on the Process:

If the attacker was able inject a malware in the receiver side which calculate the time took to decrypt the data, then the attacker will be able to deduce the key. It was a successful attack that hit during 1995 [21]. The attack cannot start directly as the receiver communicate with the legitimate transmitter as the attacker will need the exact time required to start deducing the key. Thus, it won’t affect our implementation, as till the attacker deduce the key and start decrypting data the key will be dropped, which by assuming worst case scenario the attacker might cause minor damages which won’t affect the system.

4. Lifetime of the key:

Using the same key for a long period will give the attacker higher chance of decrypting it. Thus, it is required to periodically generating keys which might affect the performance of IoT devices because of their limited computational power. Since our model requires short period of time to gather than data then drop the key then it won’t cause any problem if the attacker was able to decrypt the key later.

D. Related Work

In [3], the authors propose a convolutional neural network for indoor localization of the transmitter using RSS time-series from wireless local area network (WLAN) access points which reduces the noise and the randomness of the values presented in the RSS feeds and the last improves the accuracy. It consists of 3 steps: first, it starts by predicting the building, then, it predicts the floor number, and finally it predicts the longitude and latitude based on the building.

Moreover, it is based on 3 approaches, 1) average all the RSS vectors to input to a feedforward DNN for prediction, however, some important information will be lost due to averaging; 2) concatenate the RSS vectors into one vector to input into the feedforward DNN for prediction; and 3) build an RSS feature image and feed to the CNN, which is expected to get the most accurate results. The proposed method was evaluated and implemented on buildings where it achieved 100% building and floor prediction with an average error of 2.77 meters.

In [6], the authors proposed the use of a novel location signature CSI-MIMO (Channel State Information – Multiple Input Multiple Output) and to use the magnitude and the phase of CSI; this approach grants the CSI with the location to improve the accuracy. This approach uses both frequency and spatial diversity, and it employs both KNN and probabilistic methods while varying the test samples. In the training phase, a mobile device collects the received CSI data as a raw CSI of multiple carriers then, they generate a unique fingerprint based on the amplitude and phase for each carrier and generate the CSI-MIMO fingerprint, and store in the fingerprinting database. In the testing phase, the data collected at the unknown location is processed using the same method of the training to create its location fingerprint and to compare it with the stored fingerprints to estimate the unknown location. It uses both amplitudes and phase of (CSI-MIMO) and both KNN and maximum likelihood estimation to compare the results. The proposed method showed an improvement in the accuracy over the state-of-the-art FIFS (Fine-grained Indoor Fingerprinting System) for a KNN algorithm. In addition, accuracy of 0.95 m (less than achieved in [3]) was achieved using simple data aggregation over MIMO with optimal data.

E. Open Issues

Note that implementing machine learning algorithms at the physical layer is still a new topic. Also, PLS can leverage many properties of the hardware to enhance its performance, however, adding more attributes leads to an increase in the computational power and the complexity of the algorithm, which defeats the purpose of minimizing the error while minimizing the computational overhead. Accordingly, it is important to select just enough attributes to achieve the required accuracy.

Another challenge is the lack of available dataset to account for the large number of scenarios associated with 2 communicating devices, especially in an unstable environment, which may affect the signals by 10 db, as mentioned in [22], in addition to other issues such as Multipath Reception, Line of Sight Interference, Fresnel Zone Interference, and RF Interference.

CHAPTER III

PROPOSED METHODOLOGY

The objective of the thesis is to design security system in the PLS that relies on a ML model to authenticate the legitimate transmitter on the receiver side while exchanging packets between devices while accounting for the environment that affects the communication link. In addition, the system will cover the first handshake between both devices using a server and encryption/decryption secret keys for one time.

A. Problem Formulation and Approach

The problem can be divided into 3 parts: the first handshake between the 2 devices, continuous authentication while communicating, and the impact of the environment on the communication link.

As for the first handshake, the 2 legitimate devices do not know each other's properties, and thus, if the eavesdropper is an active listener and the 2 legitimate users tried to communicate for the first time, then the eavesdropper might intercept this communication and play the role of a middle point, which will allow her to see/edit all the messages between both users, and to compromise the sensitive data being transmitted.

Another problem is the interception of an eavesdropper of the transmitted data between both legitimate users. For example, given a transmitter, Alice, and the receiver, Bob, and an eavesdropper, Eave, when Alice and Bob are communicating, Eave is capturing the channel characteristics of Alice, and assuming that Eave has infinite computational power, she will be able to clone Alice's channel characteristics and communicate with Bob pretending to be Alice. This will cause critical problems for both legitimate parties if they are transmitting/relying on important data to continue their work.

The third problem we are tackling is the effect of the environment on the transmitted signals between both parties, as the environment cannot be assumed to be neutral, and it affects the signal which we want it to be precise for our authentication process. The environment can generate noise that affect the signal strength, which will damage some of its measurements in addition to the path losses, which must be accounted for in the authentication process.

These problems are addressed one at a time and then, the solutions are merged together to provide a secure system that can be used to enhance the security and performance of communication between legitimate parties. Thus, the approaches are divided into first handshake, model authentication and environment effect as described next.

1. *First Handshake*

Figure 5 shows the environment for the first handshake process. Alice and Bob communicate using a server, and Eave is eavesdropping on the channel. It is assumed that the server knows all users, but Alice and Bob do not know each other. Our contribution in this scenario is the use of public key technique for encryption/decryption systems, in the pre-authentication process to validate the initial identity of the transmitter and receiver and gather the PLS attributes data to train the model, by generating a secret key between both legitimate parties. ElGamal encryption technique is used due to its simplicity, speed, and security in the process. Similar to the direct key generation scenarios, the secret key will be sent directly to Alice and Bob to start communication.

The major assumption here is that the relay (server) is trusted, and the key will be received at least by Alice and Bob. After the transmission of the first few packets between Alice and Bob, the secret key will be dropped because we are assuming the worst-case scenario of Eave having high computational power. From the first few packets transmission, the receiver will be able to get measurements of the different PLS attributes, which will be used in the Model Authentication part.

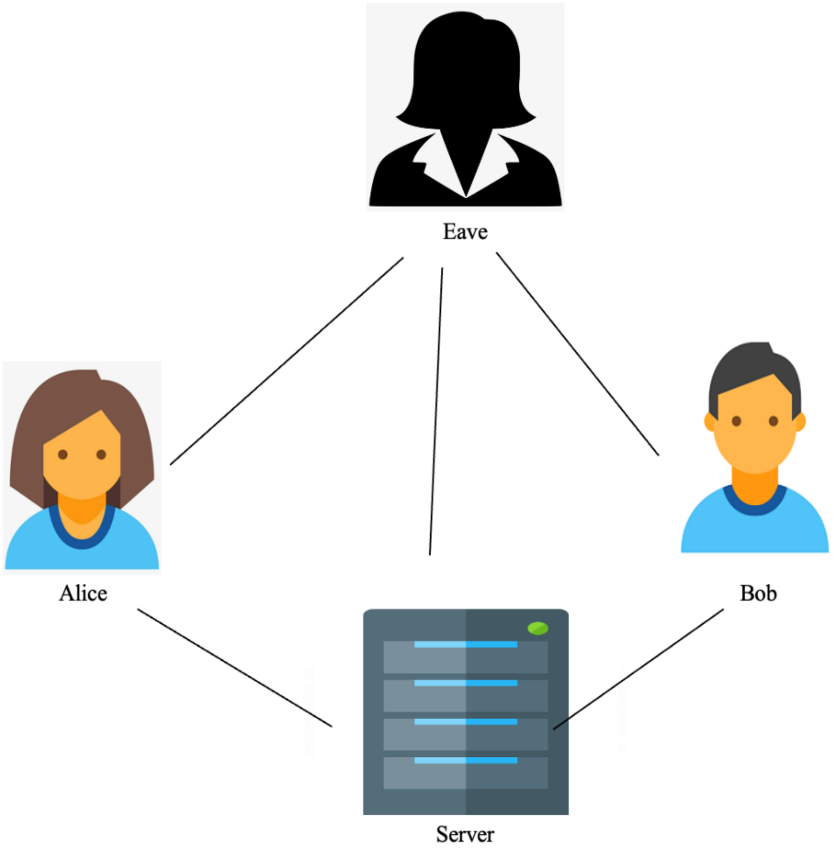


Figure 6 The Environment for the first handshake

2. *Model Authentication*

The goal is to implement a model that captures the measurements of different PLS attributes to authenticate the sender. The model will be on the receiver side, and it is not required to send nor improve any signal to the other device. In Figure 7, we will assume that Alice is the sender, Bob is the receiver, and Eave is the Eavesdropper.

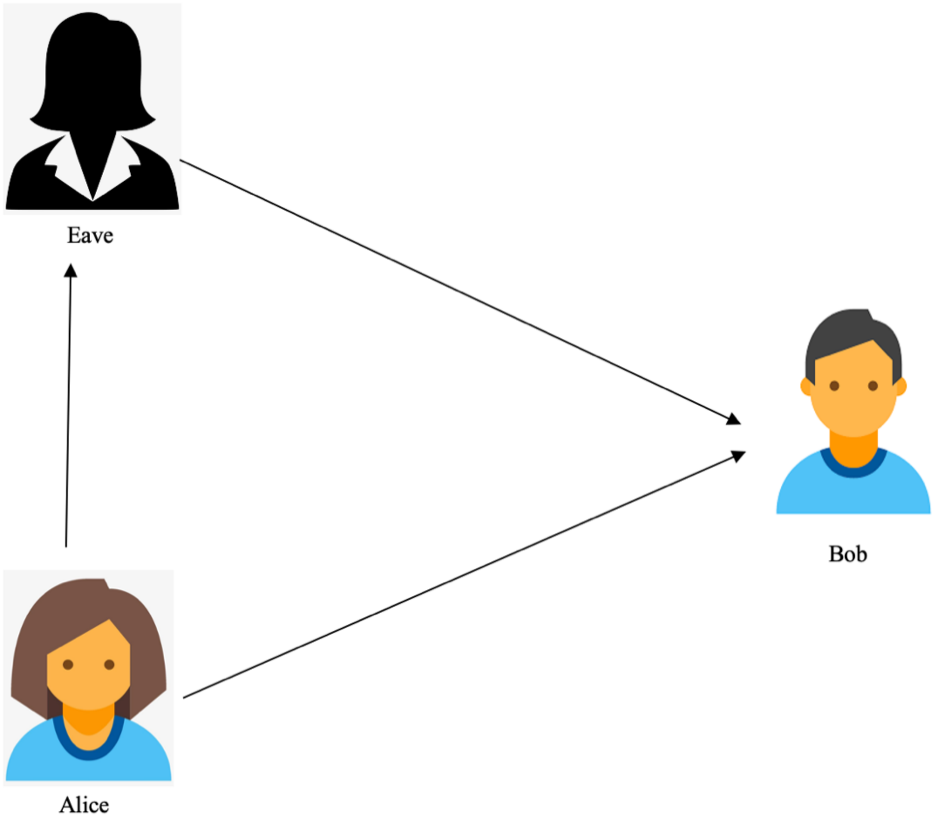


Figure 7 The continuous authentication scenario

After the Handshake, Bob will know the PLS characteristics of Alice such as RSS, CSI, and will measure the RTT, and our model will store the measurements for upcoming authentication. We will assume the worst case where Eave can clone Alice's properties and characteristics from the data transmitted to Bob. Thus, Eave will start communicating with Bob acting as Alice. Here, our model will detect changes in the measurements compared to the previous ones and it will terminate the connection.

The authentication technique followed here is comparing the current measurement with the average of the previous ones to check for the difference: if it is a slightly difference with an error of estimating the location less than 2-5 meters, as seen in previous work related to the indoor localization with error range in predicting the location after computing the location using the fingerprint, then the transmitter will be admitted and continue the communication with it, however, if the receiver detects large difference between the current measurements and the previous ones, it will check the difference range and terminate the connection for large difference or blacklist the user and take new measurements to verify before terminating the communication.

3. *Environment Effect*

Since each device will be in a different environment, this will affect the signal received and the characteristics of the communication link, or the inference of noise from different sources which will affect the signal received or the movement of the receiver or transmitter. Thus, the approach followed for this problem is to predict the change of the signal at first to minimize the range of authentication, and while communicating with the legitimate transmitter, the model will train using this data to understand the effect of the environment for better authentication accuracy.

4. *Overall System*

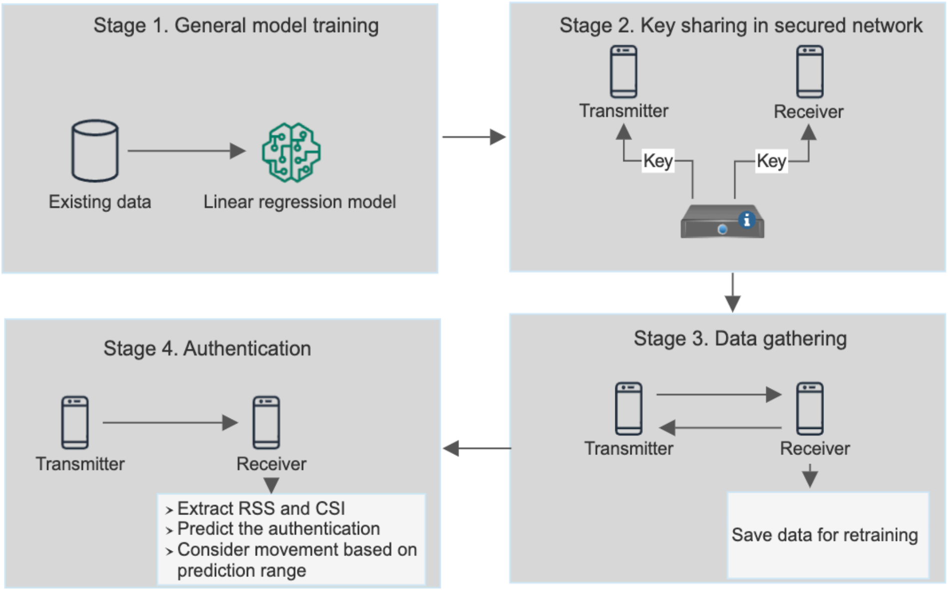


Figure 8 The overall system

Figure 8 above represents the overall system, where the first step is general training, the model then sharing the key in a secured network between Alice and Bob. Then the communication starts between Alice and Bob which will let the receiver gather the required data to retrain the model to authenticate. While authenticating, the model will take into consideration the environment and distance effects before predicting if the transmitter is authenticated or not.

Since we are targeting IoT devices, it is important to consider the movement of these devices while communicating, thus our model will take these changes into consideration by modifying the prediction range to authenticate the transmitter based on the estimated distance change per second.

In addition to considering the environment effect on the signal, specifically its effect on the RSS as mentioned above and the persistence of the CSI to these effects. Thus, the usage of CSI and RSS will avoid these effects if presented.

B. Challenges

When considering the challenges of security at the physical layer, we will consider each challenge separately (authentication, secret key distribution and environment effect prediction):

1. *Challenge 1: Limitation of the IoT devices*

Our target is to implement the model for IoT devices, which have limited computational power. Hence, the model should be simple in order not to affect the performance of these devices. In addition, these devices depend on battery life which will affect how our model will be deployed to prevent its usage all the time and drain the battery.

This challenge will be addressed by designing a simple algorithm that does not require high computational power and that will be used only while communicating with the target.

2. *Challenge 2: Secret key Management*

For the first handshake, the server needs to know each device before sending keys for the 2 legitimate parties. Thus, the server must have already communicated with the desired parties and already knows each device's ID. However, if the eavesdropper Eave intercepted any connection between the server and a legitimate device (suppose Alice) and acted as Alice for a long time without the notice of the server, the system is already hacked and implementing the model will not be effective.

3. *Challenge 3: Environment Effect*

The environment effect cannot be predicted because the weather, for example, cannot be predicted or the changes in the environment after implementing the model cannot be known. In addition to the movement of the transmitter or the receiver from an area to another with different noises that affects the communication link.

This challenge can be tackled by estimating the signals' losses by accounting for the distance between the 2 legitimate devices, which will let us minimize the range of expected received signal for the model be more accurate. In addition, the model will be trained offline periodically which will keep it in sync with the different environment changes.

CHAPTER IV

DATASET AND IMPLEMENTATION

A. Dataset preparation

The following dataset “A dataset for Wi-Fi-based human-to-human interaction recognition” [23] contains RSS and CSI data for 40 different pairs of humans moving between the transmitter and the receiver. The CSI tool [24] is used to trace the Wi-Fi signals transmitted from the Sagemcom 2704 access point. The data were collected in an indoor environment, a room of dimensions 5.3 m × 5.3 m, in the presence of furniture. Each pair were asked to perform 10 trials of 12 different human interactions at the center of the room between the 2 AP. Its goal is to advance the recognition of WIFI-based human activities in different aspects such as the usage of different algorithms to recognize the human-to-human interactions. Each pair will be located between the same transmitter and receiver and will perform different movements with each other as seen in figure 3 below where we have a receiver and legitimate transmitter, and the pair are acting between them.

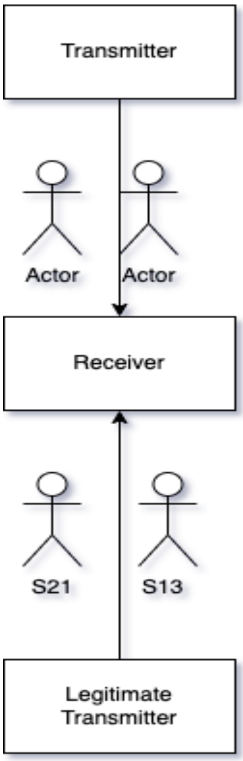


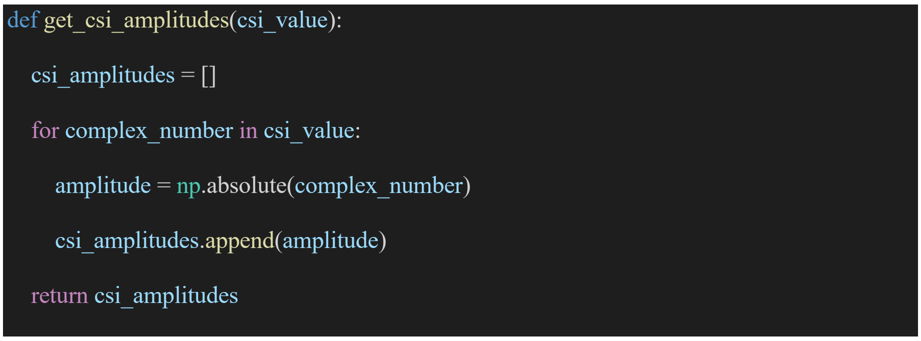
Figure 9 Scene of the dataset collection

In our implementation, to make use of this dataset, we used 4 pairs that are moving between the transmitter and the receiver (S13, S21 and any 2 others) as shown in figure 9, assuming that the transmitter is legitimate when we have S13 and S21 moving between the transmitter and the receiver. By making this assumption, our model will be trained based on the surrounding physical environment between the 2 legitimate parties, as example in the dataset used, our model will be trained based on the room and the pair between the parties, as it will be able to sense the changes of the CSI when the pair has changed. The scenario in the figure 9 can be created based on our assumptions made above. In addition, in the above scenario, the Transmitter is a duplicate of the Legitimate Transmitter (assuming worst case scenario where the unauthorized transmitter can duplicate all the properties of the authorized transmitter) as it is the same transmitter but with different pair between it and the receiver.

The features used from the dataset are RSS and CSI. The RSS consists of 1 value which is an integer. For CSI, it consists of a list of complex numbers that that depends on the antennas of the receiver and the transmitter [25], we extract the amplitude for each complex number and use it as a feature as it is used in different works [7] where the authors extracted the amplitude to create a fingerprint database to perform matching positioning at the real time. In addition, the authors in [8] relies on the amplitudes of the CSI to detect the different human activities which resulted in high accuracy in detecting different activities such as fall, and the number of people in the room. Thus, extracting the amplitudes for CSI is the most efficient way to make use of it.

The logic for preparing the data can be as follow:

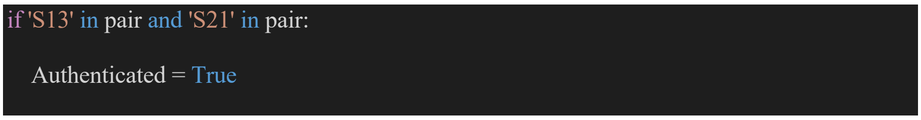
1. Extracting the RSS which is straight forward as it is an integer
2. Getting the list of CSI amplitudes from the complex numbers:



```
def get_csi_amplitudes(csi_value):
    csi_amplitudes = []
    for complex_number in csi_value:
        amplitude = np.absolute(complex_number)
        csi_amplitudes.append(amplitude)
    return csi_amplitudes
```

The function above will extract all the amplitudes for each CSI of the corresponding received signal.

3. Generating the feature to predict “Authenticated”:



```
if 'S13' in pair and 'S21' in pair:
    Authenticated = True
```

As mentioned above, the transmitter is only legitimate when we have S13 and S21 acting between it and the receiver.

To adapt to the environment's changes and in order to cover the edge cases when our model unauthorize a specific legitimate transmitter and be able to manually fix it, the model will be retrained which will affect its performance by increasing the accuracy as the data used of the specific environment will be larger.

B. Overall solution

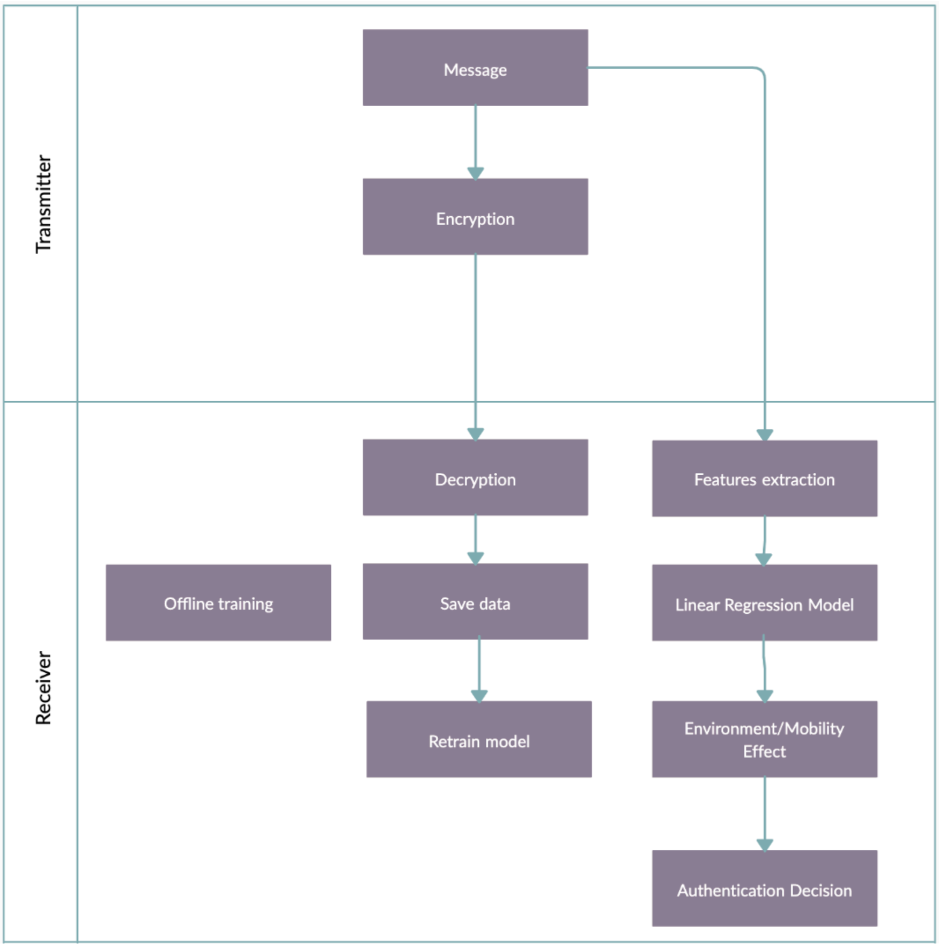


Figure 10 Overall solution

Figure 10 above shows the overall solution of the targeted problem. The first step is the offline training of the linear regression model at the receiver's end. After the offline training is done, the transmitter will start communicating with the receiver by encrypting the messages using ElGamal encryption technique and the receiver will decrypt these messages, save the data and retrain the model after having enough data. After retraining the model, the transmitter will start sending the messages directly to the receiver which will extract the required features for the linear regression model which will authenticate the transmitter based on it while accounting for the environment and movement effect.

C. Public key technique

ElGamal Encryption is asymmetric key encryption for public key technique, depends only on both the private and the secret keys generated to encrypt the data. It is based in Diffie–Hellman key exchange, and consisted of 3 components: keys generator, encryption algorithm and decryption algorithm.

The process of encryption is as follow:

1. Choose a random prime number p.
2. Choose the value of the private number a.
3. Choose the value of the generator g.
4. Get the value of the public key: $y = g^a \bmod p$
5. Share all the values except the private key a.

ElGamal encryption is simple, fast algorithm [26] which meeting the requirements of IoT devices, as it is faster than RSA in decryption of the cipher but slower in encryption. In our implementation, the main requirement is to have a fast decryption time and simple algorithm at the receiver side as it is the side where the computation and data process will take place. In addition, both algorithms are fast as the speed was compared relatively between them.

D. Model Structure

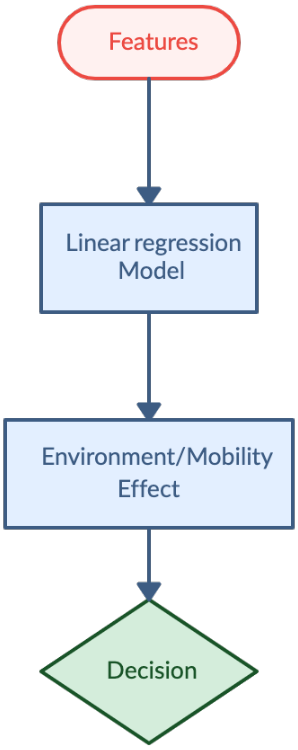


Figure 11 Model structure

Figure 11 shows the linear regression model structure where we have the features (received signal strength and channel state information) as input, then it will account for the environment and mobility effect and finally it will return the authentication decision based on these features.

E. Model authentication process

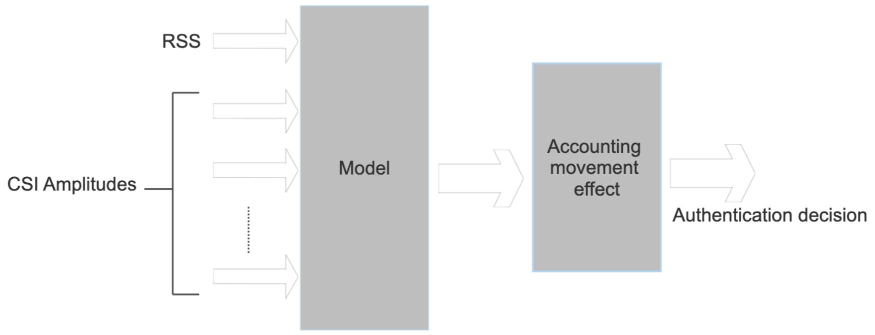


Figure 12 Authentication Process

In figure 12 above, is a process followed by our model while operating at the receiver's end, where the features are the received signal properties consisted of Received Signal Strength (RSS) and Channel State Information (CSI) ‘s amplitudes. Using these features the model will predict if the transmitter is legitimate or not. If the result is authenticated, then it will directly authenticate the transmitter and continue the communication. However, if Authenticator did not approve the transmitter, it will blacklist the last for a second chance of authentication process, if the last failed the second attempt it will terminate the connection directly.

CHAPTER V
RESULTS

A. Default classifiers results

After training our model without optimization based on the above scenario, the results obtained are as below:

Model	Accuracy	Confusion Matrix	Precision	F1-Score	Training Time	Prediction Time (whole test data)
Random Forest Classifier	0.9663	[50432, 980 2524, 50104]	0.9667	0.9663	1m 33.7s	0.9s
KNN Classifier	0.961	[50471, 941 3116, 49512]	0.9618	0.961	0.4s	5m 5.6s
Logistic Regression	0.971	[50315, 1097 1874, 50754]	0.9716	0.97144	3.4s	0.1s
Support Vector Machines	0.971	[50318, 1094 1921, 50707]	0.9711	0.971	15m 4.4s	4m 11.1s

In the table above, all models were trained and tested using the same device, same training, and testing data for fair results comparison. The models above are all classifiers and not calibrated as they used the default tuning parameters. In addition, the movement effect was not taken into consideration.

B. Metrics Used

1. Accuracy

It is the most common indicator to judge a model, it is evaluated using the below formula.

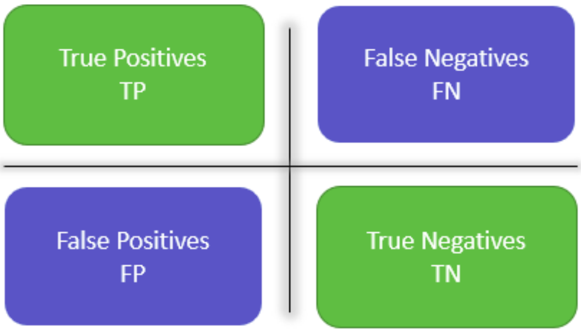
$$\frac{TP + TN}{TP + FP + TN + FN}$$

However, it is proven weak when having imbalanced classes.

In our dataset, the classes are balanced. Based on this indicator, we can notice from the table above that the most accurate models are Logistic Regression and SVM.

2. Confusion Matrix

Its output is with the same format with below table where it represents these parameters.



True Positives TP	False Negatives FN
False Positives FP	True Negatives TN

Figure 13 Confusion Matrix Table

From the table above, since our focus is on preventing the unauthenticated party to communicate with the legitimate user, **False Positive** is the important parameter for this indicator as it indicates how many times the model authenticated the unauthorized party, without losing the importance of **False Negative** as terminating the connection with a legitimate transmitter will affect the efficiency of the model. As we can notice from the table of the models that KNN Classifier performed well compared to the other models.

3. *Precision*

It can be summarized by the following: "How much the model is right when it says I am right". It is evaluated using the below formula.

$$\frac{TP}{TP + FP}$$

From the models table, we can notice that Logistic Model and SVM performed better than the other models as seen by this indicator.

4. *F1-Score*

It is the harmonic mean of 2 indicators: precision and recall. It is evaluated using the below formula.

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall}$$

As we can notice when one of these indicators is low, we will obtain a low F1-Score since the nominator is the multiplication of both indicators. In addition, both indicators are important. From the models table, we can notice that Logistic Regression and SVM performed better than the rest models.

5. Time

Since we are targeting IoT devices, our model's speed is important to take into consideration, since we want a fast model to authenticate the transmitter using limited computational devices.

Starting by the training time, Random Forest Classifier and SVM took significantly more time than the other model to train which is an important factor to take into consideration while choosing the most efficient model for our case.

The prediction time is more important in our case since our model will be predicting continuously and must stay in sync with the transmitter, from the models table we can notice KNN Classifier and SVM took a significant time to predict.

C. Model Calibration

Model calibration is the process of adjusting the model's parameters to achieve better results. These parameters are the internal configuration of the model generated based on the data. Model calibration is important as it allows each model to focus its particular probabilities for better prediction. In our implementation, we will evaluate 2 calibrated models: Logistic Regression and Random Forest.

1. Logistic regression

The model was calibrated using GridSearchCV to find the most efficient tune, the accuracy percentage varied from 97.2% and 96% where the accuracy percentage of the default model was 97.1%. However, the training time was high, few minutes (depends on the search range), compared to the default model's training time which took few seconds.

2. Random Forest

The model was calibrated using RandomizedSearchCV which generated different values for max features, max depth, minimum sample split, minimum samples leaf and bootstrap. The model achieves 97.7% for accuracy while the default model achieved 96.6%. However, the training timed for the tuned model ranged to few hours compared to the default model's training which was 2 minutes and 8 seconds.

Finally, the model calibration depends on the dataset and in our case the dataset is changing periodically. Thus, for each update of the dataset, the model will be retrained, and the training time is extremely high adding the fact of using IoT devices that has low computational power and the model's parameters varies with the dataset.

D. Movement

1. RSS

The relation between the distance between the transmitter and the receiver with the received signal strength can be expressed with logarithmic relationship. It can be expressed with Friis formula [27]. The mathematical expression of the relation is as follow:

$$RSSI = A - 10n\lg(\frac{d}{d_0}) + X_0$$

Where RSSI is received signal strength indicator, A is the signal at 1 m from the source, n is constant that ranges between 2 and 4, d is the distance between the transmitter and the receiver, d_0 is the unit distance (1m) and X_0 is the error correction term.

Thus, this equation can be used to measure the RSSI difference
at different distances (d_1 and d_2):

$$RSSI1 - RSSI2 \; = \; 10 * n * log(d_2) - \; 10 * n * log(d_1)$$

And since the human cannot move more than 4 meters per second, we can get the difference of RSSI value for 1 second range as d_2 varies between $d_1 + 4$ and d_1 - 4:

$$d1 - 4 \; \leq \; d2 \; \leq \; d1 \; + \; 4$$

$$10nlog \; (d_1 - \; 4) - \; 10nlog(d_1) \; \leq \; RSSI \; diff \; \leq \; 10nlog \; (d_1 + \; 4) - \; 10nlog(d_1)$$

$$10nlg \left(1 \; - \frac{4}{d_1}\right) \leq \; RSSI \; diff \; \leq \; 10nlg \left(1 \; + \frac{4}{d_1}\right)$$

Where d_1 is the distance between the transmitter and receiver and d_2 is the distance after movement.

2. *CSI*

WLAN protocols such as 802.11n uses Multiple Input Multiple Output (MIMO) and Orthogonal Frequency Division Multiplexing (OFDM) which enables the diversity transmission and reception of signals. Based on [28] that proposed an indoor localization based on CSI data, the CSI can be weighted as:

$$CSI_{eff} = \frac{1}{K} \sum_{k} \frac{f_k}{f_c} \times ||A||_k$$

Where CSI_{eff} is the effective CSI for estimating the distance, K is the number of subcarriers, f_k and f_c are the frequencies and A_k is the amplitude of the CSI at the kth subcarrier.

Thus, the distance between the transmitter and the receiver can be expressed as follow:

$$d = \frac{1}{4\pi}[(\frac{c}{f_0 \times |CSI_{eff}|})^2\sigma]\frac{1}{n}$$

Where d is the distance between the transmitter and the receiver, c is the radio velocity, f_0 is the frequency, n is the path loss attenuation and σ is an environment factor.

Using the above 2 equations, and calculating the relation between the distance difference and the amplitudes we concluded that the relation between the distance and the CSI amplitudes is as follow: $\frac{d1}{d2} = \frac{A_2^2}{A_1^2}$ and since the human moves 4 meters per seconds at most we can conclude:

$$\frac{d1}{d1 + 4} \le \frac{A_2^2}{A_1^2} \le \frac{d1}{d1 - 4}$$

After generating few data from the above equation to estimate the changes of RSS and CSI, the predictions of our classifier model are False (not authenticated) which is expected as the distance between the transmitter and receiver has changed. One way to solve this issue is by using a regressor instead of classifier. The results of the new data when the distance changes vary between 0.4 and 0.6 where 0 is not authenticated and

1 for authenticated. Thus, these ranges can be used to detect the changes of the distance and authenticate the transmitter.

E. Regression model results:

Model	Accuracy	Confusion Matrix	Precision	F1-Score	Training Time	Prediction Time (whole test data)
Random Forest Regressor	0.96	[50205 1207 2775 49853]	0.962	0.9617	2m 10s	0.718
KNN Regressor	0.97	[50390 1022 2568 50060]	0.965	0.965	0.175	294.88
Linear Regression	0.95		0.956	0.954	0.32	0.05

Random Forest and Linear Regression returned similar results as of classifiers. However, the KNN Regressor predicted 0 and 1 similar to the classifiers, which won't detect the distance changes.

F. Model Retraining

The model will be trained on a specific dataset to operate at first with the ElGamal decryption technique. After gathering enough data on the transmitter, the receiver will retrain the model to tune its parameters and use it only to authenticate the transmitter.

To keep our model updated with the environment changes, it will be retrained frequently to keep updates of these changes and effects. As the two legitimate parties interact with each other, the receiver will keep the required data (CSI and RSS) to retrain the model offline. The graph below shows the relation between the accuracy and the number of samples used to retrain the model.

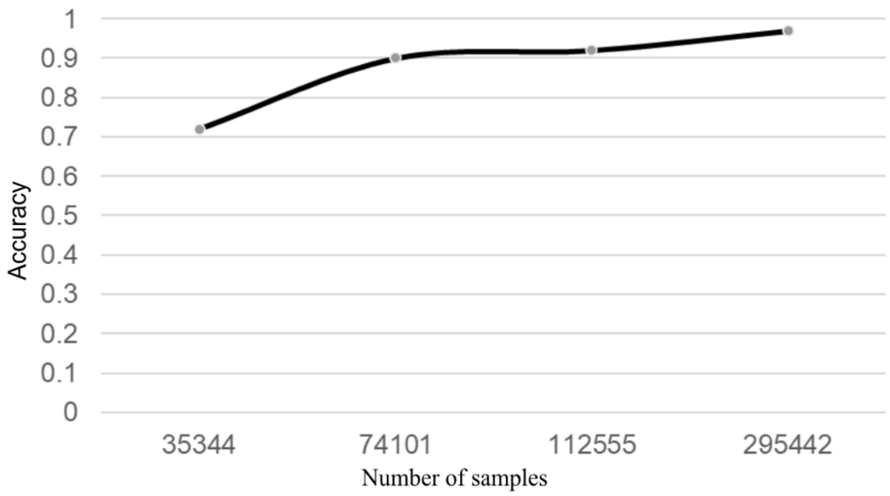


Figure 14 Graph representing the accuracy in function of the number of samples

As the number of samples data used increases, the model's accuracy increases. The graph below shows the relation between the retraining time and the number of samples used to retrain the model. As the number of samples increases the retraining time increases.

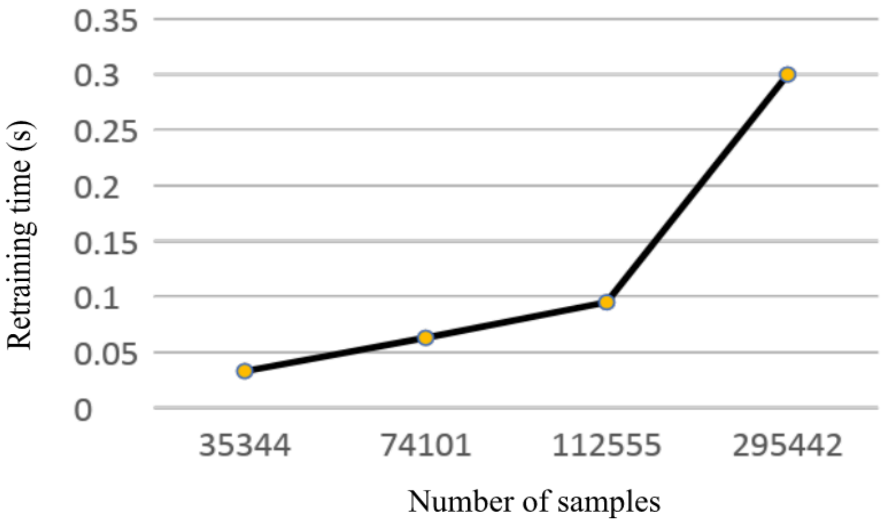


Figure 15 Graph showing the retraining time in function of the number of samples

G. Transmitter Load

The only process running at the transmitter's side is the ElGamal Encryption. It is required to encrypt the messages before sending to the receiver until the model at the receiver's side is up and running. Based on experiments done to estimate the encryption speed, it depends on mainly on the device's cores where the encryption time is ranged between 10^{-4} and 10^{-3} seconds.

H. Receiver Load

At the receiver's side, the initial training is done offline. However, the running processes are:

1. ElGamal Decryption: when the model does not have enough data to operate, the receiver will rely on the ElGamal Decryption until the model is ready to predict with high accuracy.
2. Data extraction: when the transmitter communicates with the receiver, the last will extract the required data to save/predict depending on the model's status. The following operation takes around 10^{-5} seconds.
3. Model Authentication: when the model is running, the receiver will rely on it to authenticate the receiver where the model will take 10^{-5} seconds.

Finally, for the receiver's load, the model is retrained offline to support new devices, adapting to new environment changes, etc…

CHAPTER VI

CONCLUSION

In conclusion, this thesis developed an authentication system that targets IoT devices. It relies on ML model and physical layer security attributes mainly Received Signal Strength (RSS) and Channel State information (CSI) where it relies on public key technique to secure a link to train the model (Random Forest Regressor or Linear Regressor) which considers different factors (Movement, Environment effects) while predicting. Since it targets IoT devices, the implementation is simple with fast prediction at runtime. The results verify the effectiveness of the model in the field which lays a solid foundation in the field of authentication for IoT devices.

REFERENCES

[1] M. Iwamoto, K. Ohta and J. Shikata, "Security Formalizations and Their Relationships for Encryption and Key Agreement in Information-Theoretic Cryptography," *IEEE Trans. Inf. Theory,* p. 654–685, 2018.

[2] Y. El Hajj Shehadeh, O. Alfandi and D. Hogrefe, "On Improving the Robustness of Physical-layer Key Extraction Mechanisms against Delay and Mobility," *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC),* 2012.

[3] M. Ibrahim, M. Torki and M. ElNainay, "CNN based Indoor Localization using RSS Time-Series," *2018 IEEE Symposium on Computers and Communications (ISCC),* pp. 01004-01049, 2018.

[4] F. Zafari, A. Gkelias and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys & Tutorials,* vol. 21, pp. 2568-2599, 2019.

[5] H.-X. Chen, B.-J. Hu, L. Zheng and Z. Wei, "An Accurate AoA Estimation Approach for Indoor Localization Using Commodity Wi-Fi Devices," *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC),* pp. 1-5, 2018.

[6] Y. Chapre, A. Ignjatoyic, S. A. and S. Jha, "CSI-MIMO: Indoor Wi-Fi fingerprinting system," *39th Annual IEEE Conference on Local Computer Networks,* pp. 202-209, 2014.

[7] M. Shakiba-herfeh, A. Chorti and H. Vincent Poor, "Physical Layer Security: Authentication, Integrity and Confidentiality," *Physical Layer Security,* pp. 129-150.

[8] H. Fang, X. Wang and L. Hanzo, "Learning-Aided Physical Layer Authentication as an Intelligent Process," *IEEE Transactions on Communications,* pp. 1-1, 2018.

[9] G. Biau, "Analysis of a Random Forests Model," *Journal of Machine Learning Research,* pp. 1063-1095, 2010.

[10] N. Meinshausen, "Quantile Regression Forests," *Journal of Machine Learning Research,* 2006.

[11] Y. Wang, C. Xiu, X. Zhang and D. Yang, "WiFi Indoor Localization with CSI Fingerprinting-Based Random Forest," 2018.

[12] C. Xiang, S. Zhang, Z. Zhang, S. Xu, S. Cao and V. LAU, "Robust Sub-meter Level Indoor Localization - A Logistic Regression Approach," 2019.

[13] Y. Zeng, P. Pathak and P. Mohapatra, "Analyzing Shopper's Behavior through WiFi Signals," 2015.

[14] A. Sobehy, É. Renault and P. Muhlethaler, "CSI-MIMO: K-nearest Neighbor applied to Indoor Localization," 2020.

[15] M. Ogawa and H. Munetomo, "Wi-Fi CSI-Based Outdoor Human Flow Prediction Using a Support Vector Machine," 2020.

[16] D. Sánchez-Rodríguez, M. Quintana-Suárez, C. L.-B. Itziar Alonso-González and S.-M. Javier, "Fusion of Channel State Information and Received Signal Strength for Indoor Localization Using a Single Access Point," 2020.

[17] P. Wei and Y. Zheng, "Puwen Wei and Yuliang Zheng," 2015.

[18] "tutorialspoint," [Online]. Available: https://www.tutorialspoint.com/cryptography/public_key_encryption.htm.

[19] M. Abdulla and M. Rana, "Vulnerabilities in Public Key Cryptography," 2020.

[20] D. Lazar, "Why does cryptographic software fail? A case study and open problems," 2014.

[21] E. Bresson, O. &. P. O. Chevassut, D. Pointcheval and J.-J. Quisquater, "Two Formal Views of Authenticated Group Die-Hellman Key Exchange," 2002.

[22] R. Akram, A. H. Alnakkash and O. M. Salim, "A Comprehensive Study of the Environmental Effects on WiFi Received Signal Strength: Lab Scenario," *International Conference on Applied Computing to Support Industry: Innovation and Technology,* pp. 455-464, 2019.

[23] R. Alazrai, A. Awad, B. Alsaify, M. Hababeh and M. Daoud, "A dataset for Wi-Fi-based human-to-human interaction recognition," 2020.

[24] D. Halperin, W. Hu, A. Sheth and D. Wetherall, "Tool Release: Gathering 802.11n Traces with Channel State Information. Computer Communication Review," 2011.

[25] Y. Xie and Z. Li, "Precise Power Delay Profiling with Commodity WiFi," 2015.

[26] A. P. U. Siahaan, E. Elviwani and B. Oktaviana, "Comparative Analysis of RSA and ElGamal Cryptographic Public-key Algorithms," 2018.

[27] H. T. Friis, "A Note on a Simple Transmission Formula," 1946.

[28] K. Wu, J. Xiao, Y. Yi, M. Gao and L. Ni, "FILA: Fine-grained indoor localization," 2012.