frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 07 March 2024
DOI 10.3389/fphy.2024.1357209

OPEN ACCESS

EDITED BY
Amin Ul Haq,
University of Electronic Science and Technology of China, China

REVIEWED BY
G. Palai,
Gandhi Institute For Technological Advancement, India
Riaz Ullah Khan,
University of Electronic Science and Technology of China, China
Jalaluddin Khan,
KL University, India

*CORRESPONDENCE
Ridha Ouni,
rouni@ksu.edu.sa

RECEIVED 17 December 2023
ACCEPTED 06 February 2024
PUBLISHED 07 March 2024

CITATION
Saleem K, Zinou MF, Mohammad F, Ouni R, Elhendi AZ and Almuhtadi J (2024), End-to-end security enabled intelligent remote IoT monitoring system.
Front. Phys. 12:1357209.
doi: 10.3389/fphy.2024.1357209

End-to-end security enabled intelligent remote IoT monitoring system

Kashif Saleem[1], Mohammed Farouk Zinou[1,2], Farah Mohammad[1], Ridha Ouni[2]*, Ahmed Zohier Elhendi[3] and Jalal Almuhtadi[1,4]

[1]Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh, Saudi Arabia, [2]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia, [3]Science Technology and Innovation Department, King Saud University, Riyadh, Saudi Arabia, [4]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

Introduction: Internet of things (IoT) compose of million of devices connected together over the internet. IoT plays a vital role now a days and especially in future, the most of the monitoring and data collection. The data should be secure while collection and as well in the process of transferring till the destination whether Service Organization Control (SOC) or to cloud for storage. In this paper, a secure IoT based intelligent monitoring system is proposed.

Methods: An intelligent IoT station that interacts via cellular connection to relay data to the cloud is constructed using the Waspmote platform. The algorithm is injected to automatically filter and only keep the new data for transfer to avoid redundancy. The advanced encryption standard (AES) 256-bit method is enabled for onboard data encryption and then the generated cipher text is transmitted. The encrypted data is then stored over the cloud to ensure privacy. Moreover, the mobile application (mApp) is developed to be installed on handheld devices for calling the secure data from the cloud, decrypting it, and displaying it as per user input, whether real-time or historical.

Results and Discussion: The encryption algortihm helps in securing the proposed monitoring system from brute force, man in the middle, phishing, spoofing, and denial of service (DoS) attacks. The results of the real testbed experimentation demonstrate the complexity evaluation and reliability of IoT monitoring systems with end-to-end data security in terms of encryption algorithm delay and data rate, respectively.

KEYWORDS

cloud, eHealthcare, intelligent, IoT, privacy, security, smart cities, smart grid

1 Introduction

The Internet of Things (IoT), which is widely used in daily life, is built by any device that has been enabled with internet protocol (IP) and can establish communication through the Internet. The cellular networks further boost this IoT technology that eventually enables the smart city concept [1]. The IoT is predicted to grow at an exponential rate by 2023, with approximately 30 billion linked devices. This equates to more than three gadgets for every person on Earth, and given recent events that have resulted in a more dramatic surge in online engagement, this number is likely to be substantially higher [2].The IoT has become more important in mobility, monitoring technologies, and modern data communication.

Everyday things may now connect to the Internet and communicate with each other thanks to IoT. IoT allows for the interoperability of a large number of applications and devices [3,4]. Security, throughput, large-scale connection, and ultra-reliability are all new criteria to evaluate the performance of IoT. Cellular networks play a critical role in enabling IoT by connecting and supporting communication of a large number of things via the Internet [5,6].

Several studies have investigated issues including scalability, availability, mobility, reliability, and flexibility [7]. Mobility management, interoperability of hybrid networks, and large network volumes are all design considerations for Internet Protocol version 6 (IPv6). In IoT devices, for real-time routing with IPv6, the Routing Protocols for Low-Power and Lossy Networks (RPL) were introduced to communicate over the Internet directly, without packet translation. However, because of the limited IoT node resources, an algorithm with high complexity is not a good option [8] and therefore requires some intelligent mechanisms.

Furthermore, the massive mobile interconnectivity, the IP-based open architecture, the cloud, the dynamic heterogeneity device kinds, and the diversity of the underlying access network technologies that employ information sharing and data pre-processing, together raise security concerns [5,9,10]. The immense use of these systems at Gb/s causes a lot of problems. Because of the gigantic complexity of large-scale security, data privacy and IoT deployment are the most important concerns, particularly for vital applications [11,12].

In order to provide remote monitoring with end-to-end security and user privacy, this article provides a useful IoT station-based monitoring system that is equipped with data encryption. Starting from reviewing the related literature and analyzing the issues to decide and choose the best possible options such as routing protocol, cloud setup, and encryption algorithm according to the components' limitations. Furthermore, various circumstances have distinct libraries and technical information accessible. As a result, there are several stages involved in the reconfigurations: Waspmote scripting, server configuration handling HTML pages, data transmission over UDP, and, last but not least, mApp development. The Waspmote IoT device consists of a smart city board equipped with temperature, humidity, Carbone Monoxide – CO, and Carbon dioxide – CO2 sensors. The miniaturized IoT station encrypts every communication before sending it to the cloud and gathering sensor readings in real time. The encrypted data is kept on the cloud and accessed on the portable device using an iOS native app. The main contribution of this paper includes:

- Propose a secure intelligent IoT monitoring system by integrating multiple devices and flashing them with developed program.
- Enable system with most simplest end-to-end security architecture starting from the IoT device to the user side.
- Develop a mobile application to call, decrypt, and show the real time and/or historical data.
- Conduct real-life scenario experiments to check the complete system reliability.

The proposed system could help in many different scenarios, for example, it can be linked with smart home accessories to control the air conditioner temperature to reduce energy consumption and assist the smart grid. Knowing the air quality in the user house is another benefit of the system. Moreover, it could be installed in different locations across the city to measure air quality which enables smart cities to help in multiple health situations such as people with respiratory disease. The rest of the paper is organized as follows. Section 2 presents a background of IoT applications and reviews some related work. Section 3 describes the methodology of designing a secure intelligent IoT monitoring system. The implementation and results are presented in Section 4. Finally, Section 5 concludes this paper with future work.

2 Literature review

As our world becomes more linked, new IoT applications are appearing in every industry, whether large or small. 5G also plays a crucial role in enabling new IoT capabilities [6]. We have a unique vantage point as a major provider of storage solutions from endpoints to edge devices through the core because storage is a critical component in enabling various IoT use cases. We'll go over some of the current IoT application cases in this article as follows [13].

Smart Cities: Parking, Transportation, Energy [7], and More: Creating smarter, more efficient cities is one of the most exciting IoT use cases [10]. Healthcare: Whether it is a mobile device gathering patient information at an emergency room, or an on-body monitoring system for continuous glucose, IoT devices at the edge are transforming patients' healthcare experiences. Autonomous and Connected Vehicles: Vehicles will eventually achieve Level 5 autonomy and drive autonomously without the need for human intervention. To achieve this, almost 1 terabyte of data will be stored onboard soon, with the number rising to 2+ terabytes in the next decade. Smart Agriculture: Farmers today are harnessing the potential of the Internet of Things to streamline their operations. Connected technology can track animals while they graze in open pastures as the usage of free-range livestock grows more widespread. Smart sensors can also be used in irrigation systems to save water by ensuring the proper moisture level in the soil for a certain crop [14]. AR/VR: Augmented reality (AR) and virtual reality (VR) are gaining popularity in a variety of industries, including entertainment, commerce, gaming, and medical procedures, to provide "extended reality" experiences. Wearables, Fitness Trackers, and Smart Watches: The wearables business is booming, thanks to an infusion of new personal devices.

Indeed, according to a recent analysis, the sector is predicted to reach 520.1 million units by 2025, up from 181.5 million units in 2019, representing a 19.9% CAGR over the projection period (2020–2025). Companion Robots: Companion robots are an IoT use case that has emerged in tandem with the 2020 pandemic. Proposed secure intelligent IoT monitoring system. Many of the researchers used machine learning and deep learning algorithms on the server or the where the computing is perform to make the system optimized. For example, in [15] presents a new methodology that uses state-of-the-art predictive models like Artificial Neural Network and Blockchain-based traceable mechanisms to prevent the spread of new variants of COVID-19 based on a Blockchain-based traceable model that tracks and traces. Another [16] outlines a

novel approach to statistically analyze the current state of affairs and predict COVID-19 breakouts in the future. The technique analyzes the present state of affairs in nations all around the world using weekly mobility data. To create a prediction framework, the approach is assessed using a multi-layer perceptron neural network (MLPNN), a deep learning model. Cronbach's alpha, the Case Fatality Ratio (CFR), and other measures were calculated to assess the forecasting's success.

Several approaches have been discussed in the literature for efficient monitoring. The devices have a linkage with other devices in the era of the internet of things (IoT). The prediction and monitoring systems have become more efficient and integrated with different applications to notify people robustly in real time. The connectivity with different APIs and applications requires data security, data integrity, and end-to-end privacy. The presented approaches lack data security and end-to-end security. Several approaches and techniques have been presented in this section highlighting monitoring systems [17].

In [18] an IoT-based smart environment analyzing system was presented to map the humidity, CO2 level, and temperature intensity. The data in the presented system is transmitted from the sender nodes to the receiver nodes and organized in the database. Android application and LabVIEW were utilized to monitor and share the weather information efficiently but the information sharing and data transmission did not follow the end-to-end secure channel for communication. The sleep time of the microcontroller and the power consumption of the sensors are the limitations of the proposed monitoring system.

An efficient API [19] is designed to secure the communication between applications hosted on the cloud and smart sensors using middleware secure communication. The presented approach secures communication using end-to-end security protocols by overcoming the challenges of devices like network attenuation, computation power, network buffer, and energy. Optimal scheme decider and session resumption algorithm utilized to secure the communication between devices and sensors. The session resumption algorithm resumes the device connection, disconnected due to network glitches. Supervised machine learning was employed to connect with the secure network using the optimal scheme decider method. The results of the presented technique show that secure communication can be performed robustly using the certificate and pre-shared keys. The main security concern of the proposed model is replaying checksum and collision attacks.

An advanced [20] IoT based-environment monitoring model is presented for real-time weather prediction as well as wind speed, humidity, and UV index measurement. Several sensors are utilized to collect data from the environment and data is transmitted to web pages to plot the real time graphs of weather changes. The monitoring system also comprises a monitoring app that is utilized to send alerts to people about sudden weather changes through a notification. The data collected from sensors using a web page for statistical graph plotting can be accessed anywhere using an API. The API is utilized to analyze and predict accurate predictions from past data as well as real-time weather analysis. The collected data from sensors for weather prediction can be utilized for future prediction due to compact data size. The present monitoring system did not comply with end-to-end data privacy and security which is a drawback of the implemented system.

A wireless sensor network (WSN) [21] was utilized to design an IoT-based weather monitoring system. The main goal of the presented approach is to monitor the weather in remote areas and provide access to the data collected in these areas through the Internet. The presented model contains two types of nodes. One is for information extractions with access through webpages anywhere on the internet and the other sends alerts about the harshness of weather to people when the parameter of data collected for weather prediction exceeds the threshold limit. The proposed system collects data and provides access over the internet but does not deal with the data security required for data access and data transmission. IoT systems normally are based on wireless network sensors (WSN). WSNs are utilized for the weather monitoring system to measure and predict the weather conditions and data is transmitted over the network.

The article [22] stress the significance of patient data security and privacy in IoT-enabled healthcare systems. The authors provide an IoT sensor-enabled medical healthcare (SMSH) system's safe surveillance technique. The suggested method entails keyframe picture encryption and intelligently recorded video summaries onto the server. First, a well-organized keyframe extraction process is triggered by the visual sensor to extract relevant picture frames, after which an alarm is delivered to the appropriate authority within the healthcare system. Second, the ultimate determination of what transpired with the keyframes that were taken is kept secure from any hackers. In [23] the authors conduct further research by using cosine-transform encryption to make it more secure from any adversary.

A novel WSN-based weather monitoring is introduced in [24]. The implemented system collects data from the environment of different conditions like air, humidity, heat and cold and stores all the data on the cloud-based storage system. The stored data can be provided to a single client or multiple users on their smartphones when people want to know the weather conditions. The presented WSN-based system has been implemented in real time to evaluate the adaptability, sustainability, and scalability of the weather monitoring architecture. The limitation of the proposed model includes power availability in remote areas which can be improved using solar panels in the wild.

The authors in [25] offers a process for IIoT ecosystem validating efficacy at every step. The technique is built on perceptively extracted keyframes, gorgeous monitoring, and lightweight cosine functions processed by hybrid approach chaotic map keyframe image encryption. Compared to the earlier keyframes image encryption approach, the generated result has a shorter execution time, is more resilient, and can be implemented at a reasonable cost while maintaining security [23].

In [26], the authors introduce a class of smart electronic gadget for automotive applications that manages an anti-theft IoTs security system. The gadget contributes significantly to the development of the smart city framework and is intended to reduce the workload for security officers. The way the gadget operates is to prevent unauthorized individuals from accessing the car until an authorized password is provided or an SMS is delivered to the car. The gadget may also be used to remotely operate the vehicle's engine-stopping system. It has been demonstrated that the produced prototype is both affordable and appropriate for real-world use. An effective IoT alert

TABLE 1 Realted work comparison.

References	Hardware						Sensors			
	Mote	Connectivity	Algorithm	Cloud	Encryption	App	Temp	Hum	CM	CD
[22]	-	-	YOLOv3 MATLAB Sim based	-	CTC-IES 256	-	-	-	-	-
[21]	Arduino Uno	WiFi	Exp	✓	-	✓	✓	✓	-	✓
[16]	-	-	YOLOv3 MATLAB Sim based	-	STC-IES 256	-	-	-	-	-
[25]	-	-	MLPNN - Sim based	-	-	-	-	-	-	-
[24]	Arduino Uno	XBee Pro + 2G Cellular	Exp	✓	-	✓	-	✓	✓	✓
[27]	-	-	ACO Sim based	✓	-	-	-	✓	-	-
Proposed Design	Waspmote	WiFi + 3G Cellular	Exp	✓	AES 256	iOS	✓	✓	✓	✓

where: MLPNN, Multi-layer Perceptron Neural Network; Sim, Simulation; Exp, Real Testbed Experimentation; CTC, Cosine-Transform-based Chaotic Sequence; STC, sine tent cosine; IES, Image Encryption System; ACO, Ant Colony Optimization; Temp, Temperature Sensor; Hum, Humidity Sensor; CM, Carbone Monoxide Sensor; CD, Carbone Dioxide Sensor.

system based on MQTT protocol was introduced that enables compact and quick communication. To sum up, the article presents a novel device that is efficient, cost-effective, and suitable for practical implementation, and it is a significant contribution to the field of automobile security.

In [27] presents a new kind of monitoring system that is based on an ant colony optimization algorithm. The system is designed to monitor the rural environment and provide real-time data to the users. The system is composed of three parts: the sensor network, the data processing unit, and the user interface. The sensor network is responsible for collecting data from the environment, while the data processing unit processes the data and sends it to the user interface. The user interface is responsible for displaying the data to the user in a user-friendly way. The ant colony optimization algorithm is used to optimize the routing of the data from the sensor network to the data processing unit. The algorithm is designed to minimize the energy consumption of the sensor nodes and to maximize the lifetime of the network. The experimental results show that the proposed system is efficient and effective in monitoring the rural environment. The summarized comparison of the related work is presented in Table 1.

3 Methodology

Since IoT is quickly growing, the number of connected devices is increasing and getting more powerful. There are lots of devices that could be used as an IoT development platform, including Arduino Genuino UNO, Raspberry Pi 3, WeIO, BeagleBone, and Nanode. Each one has its advantages and disadvantages. One of these platforms, Waspmote has the most battery life that lasts from 1 to 5 years depending on the application. Moreover, it simplifies the hardware connection because it has an API to deal with different parts of the platform modules rather than interact directly with the pins [28].

Waspmote is a sensor gadget that may be used to create IoT projects. The IoT hardware architecture has been specifically engineered to operate at exceptionally low power consumption. Any of the sensor interfaces, as well as the radio modules, can be turned on and off using digital switches. Waspmote is the lowest consumption IoT platform on the market (7 μA) thanks to three different sleep modes [29] Figure 1 shows the hardware layout for Waspmote.

Another major feature of Waspmote is over the air programming, in recent years, the notion of Wireless Programming, also known as Programming Over the Air (OTAP), has been utilized to reprogram mobile devices such as cell phones. With the new concepts of IoT, M2M, and the Wireless Sensor Networks where networks consist of hundreds or thousands of nodes, OTAP is taken in a new direction, and for the first time it is used with both licenses such as 5G and unlicensed protocols such as WiFi [29,30]. The main benefits of OTAP are:

- Allow for firmware upgrades or changes without requiring physical access;
- The latest firmware is installed by requesting an FTP server, which helps to preserve battery life;
- Upgrade an entire network in a matter of minutes.

In order to implement the project, the following hardware parts are required:

- 1x Waspmote 802.15.4 uFL;
- 1x Waspmote Gas Sensors Board v20;
- 1x Temperature Sensor;
- 1x Humidity Sensor;
- 1x Carbone Monoxide Sensor;
- 1x Carbone Dioxide Sensor;
- 1x 2300 mAh LiPo Battery;
- 1x miniUSB Cable;
- 1x Cellular Module.

The required software are:

- Waspmote IDE (https://development.libelium.com/waspmote-ide-v06/download-ide-windows).

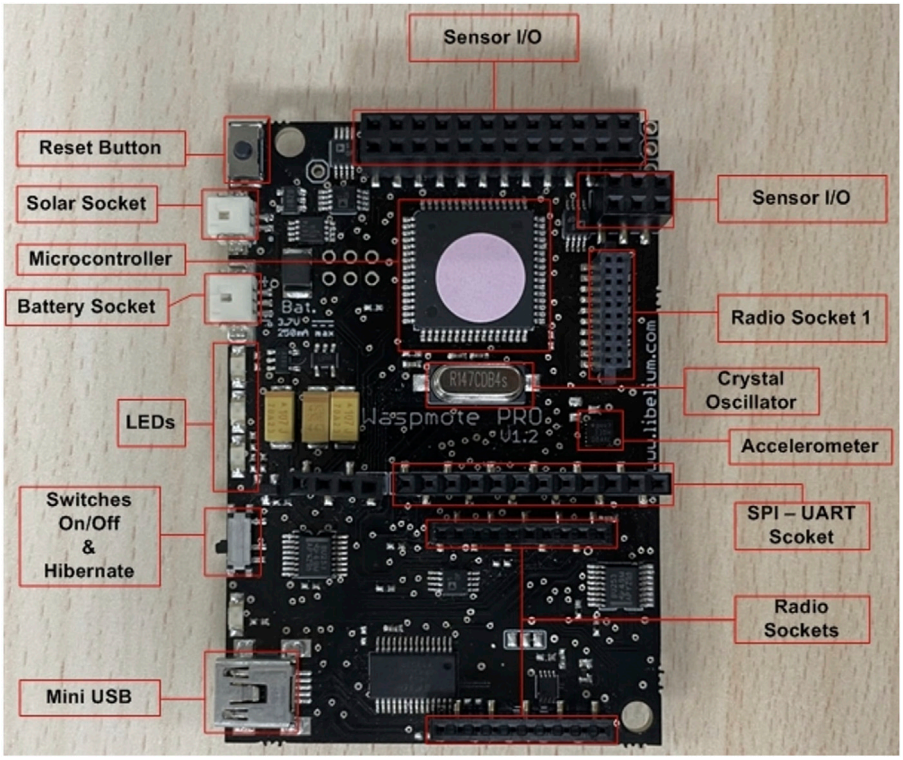

FIGURE 1
Waspmote layout.

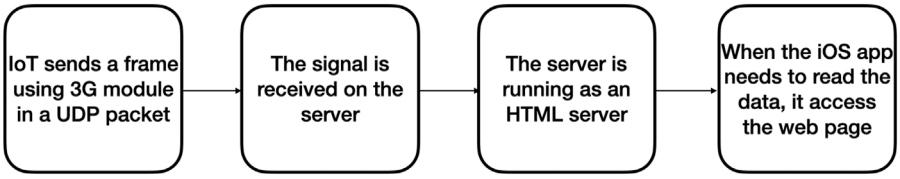

FIGURE 2
Data flow.

- Waspmote API v32 (http://downloads.libelium.com/waspmote-pro-api-v032.zip).

Note that downloading the IDE will come with the latest API version, but it is needed to downgrade into v32 because the Gas Sensors Board v20 requires an API version of 32.

Networking plays a main role in IoT where any IoT device needs to communicate to perform the purpose that was intended to serve.

Several possibilities could be used to connect with the Waspmote IoT platform. We list them below:

- Connect using a WiFi module.
- Connect using a Xbee module.
- Connect using a Cellular module.

The first two options limit the connection to the local area network (LAN). To connect and transfer the IoT measurements over the internet, the Cellular module is used. This module is enabled with 3G (Third Generation) cellular networks that operates on specific frequency bands. The 3G network primarily uses the UMTS (Universal Mobile Telecommunications System) technology for which the frequency range is 2100 MHz band 12 and the data transfer rates of up to 2 Mbps. In two ways mainly 3G dealt with the high-frequency noise and interference, firstly 3G base stations use smart antennas (such as MIMO) to enhance signal reception and reduce interference. These antennas dynamically adjust their beam patterns to focus on specific users or areas. Secondly, 3G employs robust error correction techniques that detect and correct errors in transmitted data, ensuring reliable communication. In the next step, the signal is generated from the IoT device towards the end user's mobile phone to view it in the developed mobile iOS application.

The flowchart, given in Figure 2, shows how the data is transferred.

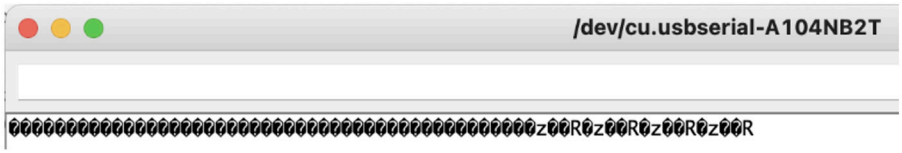

FIGURE 3
Message not recognized by IDE terminal.

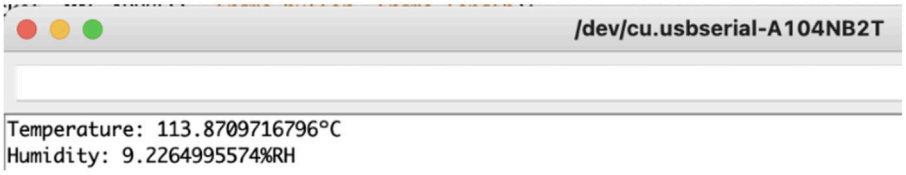

FIGURE 4
Message recognized by IDE terminal.

Yet, the data is transferred as plain text, and if someone accesses the website or hijacks the forwarded packets using a man-in-a-middle attack, the hacker can easily monitor or even manipulate the actual data or even launch other attacks [11]. Therefore, the end-to-end security is implemented from the onboard frame encryption based on an advanced encryption standard (AES) algorithm with a key length of 256 bits [31] to ensure information authenticity and confidentiality. The onboard AES algorithm encrypts the message m based on electronic codebook (ECB) cipher mode with ZERO padding at Waspmote to generate cipher c as given in (1) [32].

$$c_n = e(k, m_n) \tag{1}$$

Where e represents the encryption process, k is the predefined key, and n is the sequence of the plaintext blocks from the beginning till the end.

The c is transferred to the given destination over the cloud as per configuration for storage in an encrypted form to maintain data security and privacy. The user can call the data from the cloud on the handheld device through the mApp. The data c transfers from the cloud to mApp is still encrypted, until and unless the user inputs the given key for decryption d. The real-time data is called, decrypted with k based on (2) [32], and displayed as soon as the c reaches the server from the IoT station. In addition to the real-time data, the history can be viewed by the user based on the duration entered by the user. The history data also reaches the user side in encrypted form c and is displayed on mApp; after getting d by the mApp when a user unlocks it by k.

$$m_n = d(k, c_n) \tag{2}$$

Furthermore, the Waspmote IoT station is enabled with an intelligent mechanism to check and prevent messages with similar environmental values. The mote periodically senses the environmental values and tally with the previous memorized record. If the values are similar, the mote copies the new values in the memory and erases the old record, locally. Otherwise, the message is generated, encrypted, and forwarded with only the new values. In this way, frequent messaging can be avoided to help minimise massive network traffic and especially traffic overload. Secondly, the reduction in communication maximizes the battery lifetime.

On the security side, the intelligent mechanism makes it very hard for a cryptanalyst to exploit the irregular communication and as well to analyze the similarities due to the newer value in each transmission.

4 Implementation and results

We have started by installing the Waspmote platform IDE (Integrated Development Environment) on our computer. It is needed to connect and deploy code into the hardware itself through a significant capability in supporting Windows, macOS, and even Linux. After making the hardware connections and installing the battery, we run the default program that comes with the hardware. It reports every 5 s the battery level, MAC address, and the temperature of the board. However, the IDE terminal was not able to recognize the message as shown in Figure 3.

The baud rate should be set to 115200 instead of 9600 because this value is used by Waspmote v1.2. Now, the message is decoded and shown correctly in Figure 4. The next step is to install the Waspmote Smart Cities Board to equip the required sensors. Then, we wrote and installed the program which reads both humidity and temperature, and the output is presented in Figure 4.

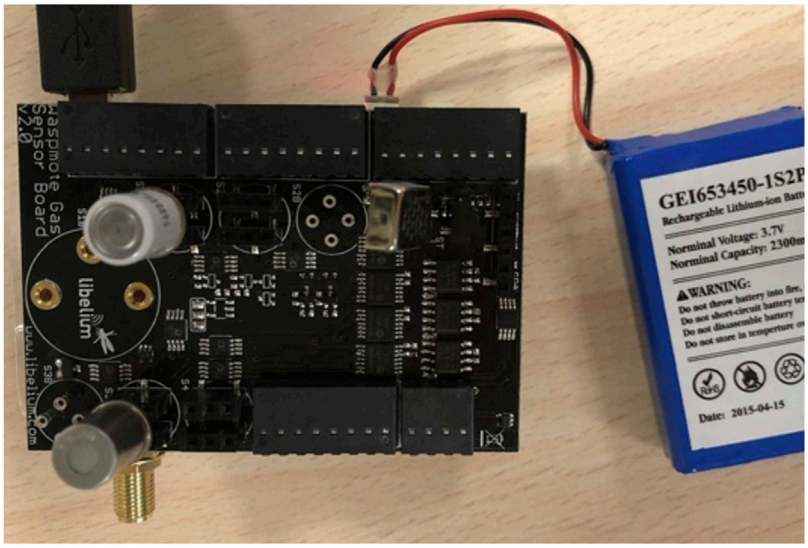

FIGURE 5
Fully equipped smart sensor board for Waspmote.

```
Current ASCII Frame:
Length: 85
Frame Type:  128
frame (HEX): 3C3D3E800523343033343537383439234761737365732331234750 5F54433A2D34392E3033322347505F48554D3A312E37372347505F43
frame (STR): <=>@ #403457849#Gasses#1#GP_TC:-49.032#GP_HUM:1.77#GP_CO2:333.333#GP_CO:0.939#BAT:32#
```

FIGURE 6
The code output.

The next step consisted of deploying the rest of the sensors, which are CO2 (Carbon Dioxide) and CO (Carbon Monoxide) sensors. Figure 5 shows the fully equipped Smart Cities Board.

However, when working on the code that reads all the sensor readings from the board, some major errors appeared and the code was not compiling at all. After investigation, we detected that the signal names were turned out (renamed) in the Interrupt Vector Table. Unfortunately, the given solution did not solve the issue and introduced a new range of errors! After multiple searches, the problem was identified. The Smart Sensors City Board requires an API of version 32, as mentioned in Section 3, which leads to a downgrade of the API. Finally, the problem is resolved and can compile the code and read the sensor values as shown in Figure 6.

4.1 Networking

The Cellular module configuration is set up in the first step including the APN, username, password, and PIN for the sim. Then, the Waspmote can send user datagram protocol (UDP) and/or transmission control protocol (TCP) packets. Due to the traffic overhead issues and other reasons as listed below the UDP is enabled.

- UDP packet is much simpler than TCP.
- UDP is a connectionless protocol, so we do not need to deal with establishing a connection process.
- According to the selected application, the packet loss does not matter, but the massive traffic overhead can reduce the lifetime of the device.

In the second step, cloud service is availed with the information as given in Figure 7.

After running the Ubuntu Linux distribution, the server is connected by using a secure shell protocol (SSH), and then a UDP connection is configured. The firewall settings are configured in the system according to the traffic requirements. The UDP on port 6000 is opened by entering the following command:

sudo ufw allow 6000/udp.

The firewall is then restarted to update the settings:

sudo ufw disable.

sudo ufw enable.

The UDP part is completed at this point. The third step in the experimentation consists of running the server as an HTML server, which is achieved through an Apache server using the following commands:

sudo apt install apache2.

FIGURE 7
Cloud Configuration.

FIGURE 8
Single record data stored on the cloud and displayed on the HTML page.

FIGURE 9
The ciphertext transferred and stored on the cloud.

systemctl enable apache2.

systemctl start apache2.

The server is now configured to run as an HTML web server, that can be accessed directly by entering the server IP in the web browser. One last step to enable the server to receive UDP packets and write their content directly to the HTML page is enabled by entering these commands.

First, the directory is changed to the path where the Apache server is running:

cd/var/www/html

Second, UDP packets are captured by using NetCat tool:

nc -u -l 6000 -k > index.html

where

nc: the command to use NetCat tool.

-u: use UDP.

-l: listening mode.

6000: is the port number to allow the UDP traffic early in the firewall.

-k: keep the connection alive. This command is very important to keep the tool continuously listening. Otherwise, the tool will stop listening after receiving the first packet and needs to enter the command manually multiple times.

> index.html: write the received content into the webpage defined by the directory that we are running the Apache web server.

Figure 8 shows the received data from the IoT device and is displayed on the HTML page.

4.2 Encryption

In order to protect the data, encryption has been enabled in the intelligent monitoring system before sending the data to ensure secure transfer and storage. This is one of the reasons for choosing the Waspmote platform as it supports encrypting frames before sending. Therefore, the AES 256 encryption algorithm is included for the secure transfer as shown in the generated ciphertext in Figure 9.

```
struct Frame {

    let sequence: Int
    let temperature: Double
    let humidity: Double
    let CO2: Double
    let CO: Double
    let battery: Int

    init(data: String) {

        let arrayData = data.components(separatedBy: "#")

        self.sequence = Int(arrayData[3])!
        self.temperature = Double(arrayData[4].components(separatedBy: ":")[1])!
        self.humidity = Double(arrayData[5].components(separatedBy: ":")[1])!
        self.CO2 = Double(arrayData[6].components(separatedBy: ":")[1])!
        self.CO = Double(arrayData[7].components(separatedBy: ":")[1])!
        self.battery = Int(arrayData[8].components(separatedBy: ":")[1])!
```

FIGURE 10
Frame structure.

```
func decryptData(_ string: String, key: String) -> String? {

    let bytes = Array<UInt8>(hex: string.filter({$0 != "\n"}))

    do {
        // decrypt
        let decryptedDataArray = try AES(key: Array("\(key)".utf8), blockMode: ECB(), padding:
            .zeroPadding).decrypt(bytes)

        let string = String(bytes: decryptedDataArray, encoding: .ascii)

        let stringRemovedZeroPadding = string!.filter { char in
            char != "\0"
        }
        return stringRemovedZeroPadding

    } catch {
        print(error)
        return nil
```

FIGURE 11
Decryption function.

4.3 Mobile application and data decryption

The native iOS application is built to enable the user to discover the environmental readings globally in real-time, which were sent from the IoT device over the internet.

Every message "send", as shown in Figure 8 from the IoT device, begins with <=> using # separator. This information is utilized to create an object called a frame to map the data in the app. Figure 10 shows the frame structure declaration and initialization.

In the initialization process, each frame is mapped to the corresponding value that it represents. For example, the 3rd element in the original frame represents the frame sequence number, the 4th element represents the temperature sensor value, and the 5th element represents the sensor humidity value.

The encrypted data needs to be decrypted for further analysis or viewing. The decryption is performed by using the function inside the app as shown in Figure 11.

Furthermore, the rest of the app code uses the user interface (UI) and manages different states of the app, including data encryption, decryption, or in case the key is wrong. The screenshot of the sensed data output along with the mobile application coding is shown in Figure 12. In addition, a video recording of the app in action can be accessed via https://youtu.be/Vt5Jz_cZ_xk.

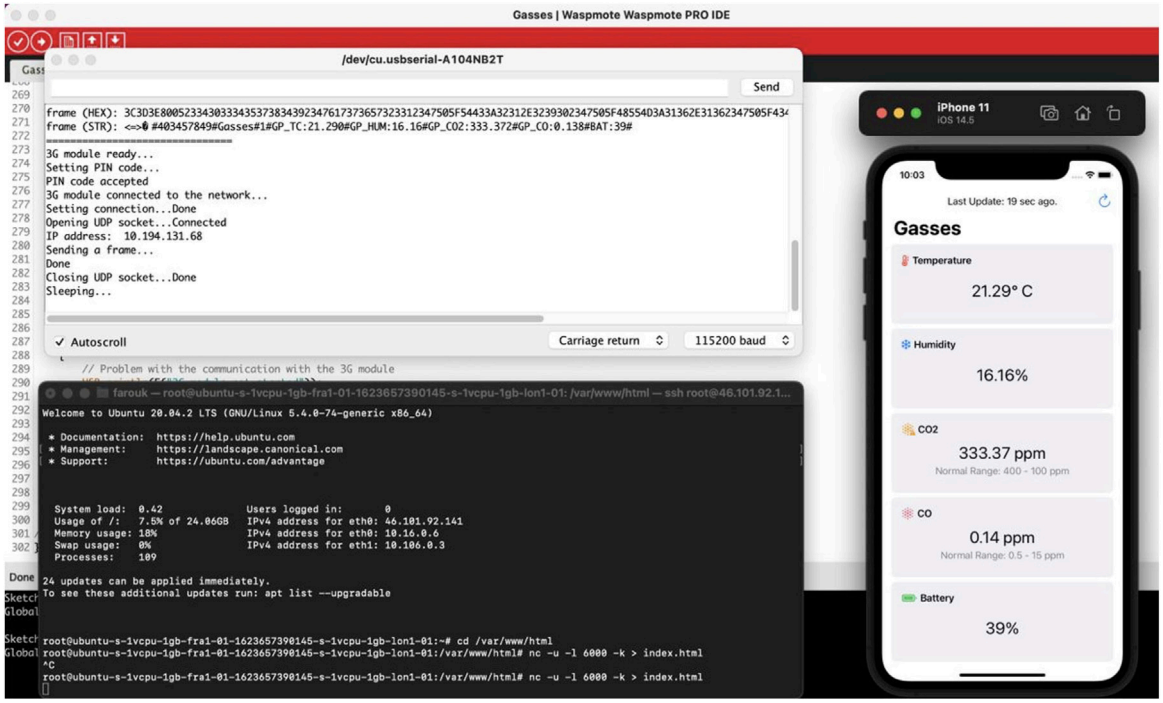

FIGURE 12
The sensed data output with mobile application coding.

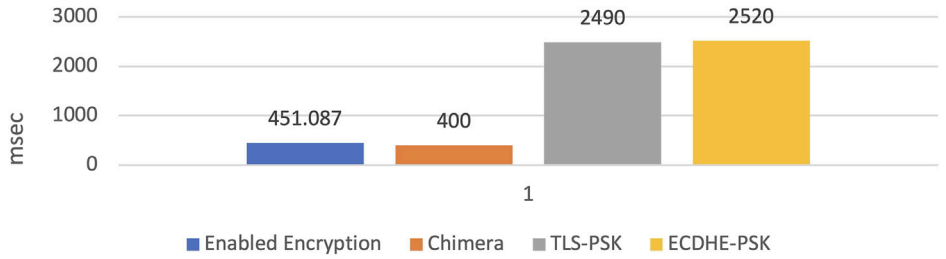

FIGURE 13
Delay comparison.

4.4 Result analysis

The proposed IoT based monitoring system has been experimented with in real time. In the first scenario, both normal data and encrypted data are collected locally through a WLAN connection for about 96 h. Figure 13 shows the delay (in ms) given for various encryption algorithms. The average encryption delay at the node from the collected 617 records is 451.1 msec.

In the second scenario, the IoT Waspmote is configured to send encrypted data over a cellular network and store it on the cloud. The ciphertext is then called on the mobile side by the developed iOS application. The legitimate user can access the data by entering the correct passcode. The process on the backend starts when the application calls the ciphertext and decrypts it based on the passcode entered by the user. In this scenario, the experiment run four times, in first the packet is generated every 5 s and sent as normal open data. In the second, the packet is generated every 50 s and sent as normal data. In the third, the packet is generated every 5 s but the Waspmote encrypts the data and sends it as ciphertext. In the last experimentation, the packet is generated every 50 s and sent as ciphertext. When data is transferred to the cloud, the configured droplet on the configured cloud shows the data rate of transferred ciphertext with 5 milliseconds packet rate shown in Figure 14 and with 50 milliseconds in Figure 15. While observing Figure 14, the high bandwidth of about 2.7 kb/s can be seen, because of the high packet rate generated on the Waspmote side. Figure 15 shows very little bandwidth of about 48 bits per second in front of Figure 14, due to the packets generated by Waspmote with long duration.

FIGURE 14
Cloud receiving ciphertext with high bandwidth.

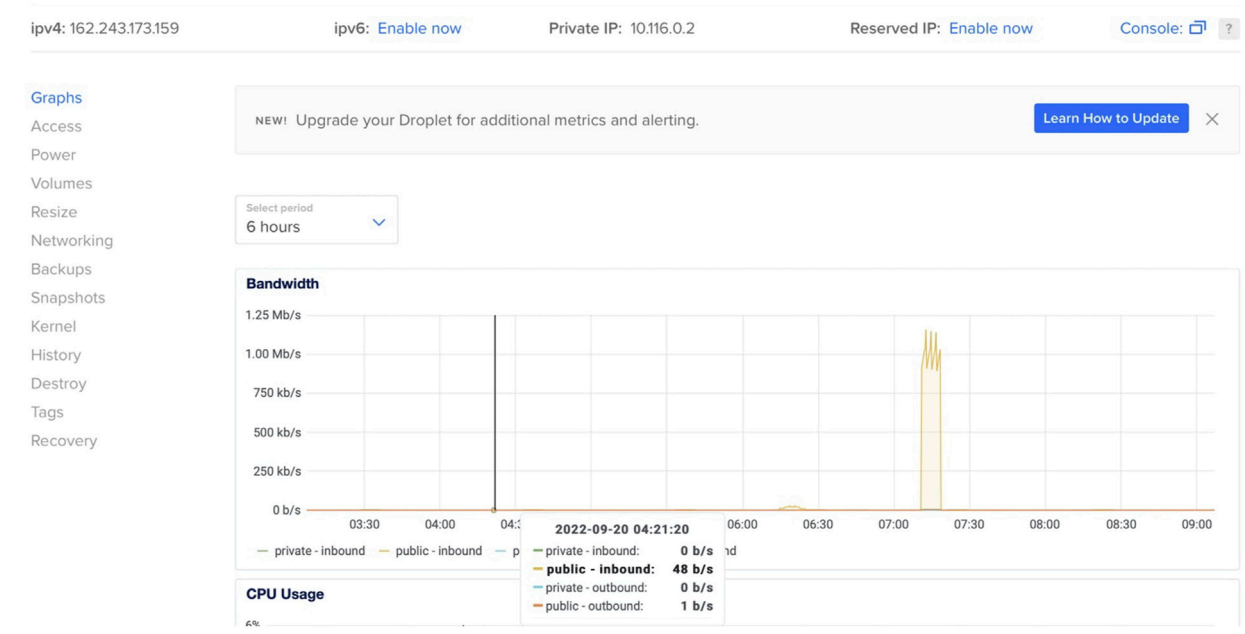

FIGURE 15
Cloud receiving Ciphertext with Less Bandwidth.

5 Conclusion

The intelligent monitoring secure IoT station has been proposed, implemented and tested experimentally in this research paper. The related literature is reviewed and according to the most recent IoT requirements, the important privacy and security issues in remote sensing are addressed. The device has been coded and transformed into a complete secure remote monitoring system along with a developed mApp to fetch data from the cloud in a secure manner. The Waspmote based IoT station is intelligent in a manner to eliminate frequent communication and data packet redundancy. End-to-end security is enabled, starting from the onboard data encryption on the IoT station. The testbed of the intelligent IoT station is experimentally tested which shows the performance with the high data rate of up to 200 packets per second that comes up with a bandwidth of 2.7kilobit per second.In future, multiple IoT remote monitoring stations will be programmed and deployed to check the

compatibility and performance. The encrytion algorithm will be enhanced with better security because AES 256 is not guaranteed against quantum attacks. Moreover, the system will be enabled with mobile monitoring nodes to enable eagle eye viewing and as well better connectivity between multiple nodes especially in the case of multi hop communication.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

KS: Conceptualization, Formal Analysis, Methodology, Supervision, Visualization, Writing–original draft, Writing–review and editing, Funding acquisition, Project administration, Resources, Software, Validation. MFZ: Conceptualization, Software, Validation, Visualization, Writing–original draft, Data curation. FM: Conceptualization, Visualization, Formal Analysis, Methodology, Supervision, Writing–review and editing. RO: Conceptualization, Formal Analysis, Methodology, Supervision, Visualization, Writing–review and editing, Writing–original draft. AE: Formal Analysis, Investigation, Methodology, Validation, Visualization, Writing–review and editing. JA: Funding acquisition, Investigation, Project administration, Supervision, Visualization, Writing–review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by the Deputyship for Research & Innovation, "Ministry of Education" in Saudi Arabia for funding this research work through the project number (IFKSUDR_D104).

Acknowledgments

The authors extend their appreciation to the Deputyship for Research & Innovation, "Ministry of Education" in Saudi Arabia for funding this research work through the project number (IFKSUDR_D104).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Zhao Y, Li S, Chen H, Xu Y Application of smart city construction in a new data environment. *Front Energ Res* (2022) 10. doi:10.3389/fenrg.2022.908338

2. Cisco. *Cisco annual internet report (2018–2023) white paper*. San Jose, CA, USA: Cisco (2020).

3. Bayılmış C, Ebleme MA, Küçük K, Sevin A A survey on communication protocols and performance evaluations for internet of things. *Digital Commun Networks* (2022) 8: 1094–104. doi:10.1016/j.dcan.2022.03.013

4. Vilela PH, Rodrigues JJPC, Solic P, Saleem K, Furtado V Performance evaluation of a fog-assisted iot solution for e-health applications. *Future Generation Comp Syst* (2019) 97:379–86. doi:10.1016/j.future.2019.02.055

5. Saleem K, Alabduljabbar GM, Alrowais N, Al-Muhtadi J, Imran M, Rodrigues JJPC Bio-inspired network security for 5g-enabled iot applications. *IEEE Access* (2020) 8: 229152–60. doi:10.1109/ACCESS.2020.3046325

6. Shen Y, He T, Wang Q, Zhang J, Wang Y Secure transmission and intelligent analysis of demand-side data in smart grids: a 5g nb-iot framework. *Front Energ Res* (2022) 10. doi:10.3389/fenrg.2022.892066

7. Gu Q, Qu Q Towards an internet of energy for smart and distributed generation: applications, strategies, and challenges. *J Comput Des Eng* (2022) 9:1789–816. doi:10.1093/jcde/qwac087

8. Saleem K, Chaudhry J, Orgun MA, Al-Muhtadi J A bio-inspired secure ipv6 communication protocol for internet of things. In: 2017 Eleventh International Conference on Sensing Technology (ICST); December, 2017; Sydney, NSW, Australia (2017). p. 1–6. doi:10.1109/ICSensT.2017.8304428

9. Snow S, Happa J, Horrocks N, Glencross M Using design thinking to understand cyber attack surfaces of future smart grids. *Front Energ Res* (2020) 8. doi:10.3389/fenrg.2020.591999

10. IbneÂ Hossain NU, Nagahi M, Jaradat R, Shah C, Buchanan R, Hamilton M Modeling and assessing cyber resilience of smart grid using bayesian network-based approach: a system of systems problem. *J Comput Des Eng* (2020) 7:352–66. doi:10.1093/jcde/qwaa029

11. Stergiou C, Psannis KE, Kim B-G, Gupta B Secure integration of iot and cloud computing. *Future Generation Comp Syst* (2018) 78:964–75. doi:10.1016/j.future.2016.11.031

12. Yaseen M, Saleem K, Orgun MA, Derhab A, Abbas H, Al-Muhtadi J, et al. Secure sensors data acquisition and communication protection in ehealthcare: review on the state of the art. *Telematics Inform* (2018) 35:702–26. doi:10.1016/j.tele.2017.08.005

13. Iarovici Y *Top 10 IoT use cases. Report* (2022).

14. Ma B, Wang Q, Xue B, Hou Z, Jiang Y, Cai W Application of uav remote sensing in monitoring water use efficiency and biomass of cotton plants adjacent to shelterbelt. *Front Plant Sci* (2022) 13:894172. doi:10.3389/fpls.2022.894172

15. Khan RU, Haq AU, Hussain SM, Ullah S, Almakdi S, Kumar R, et al. Analyzing and battling the emerging variants of covid-19 using artificial neural network and blockchain. In: 2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP); December, 2021; Chengdu, China (2021). p. 101–5. doi:10.1109/ICCWAMTIP53232.2021.9674142

16. Khan RU, Almakdi S, Alshehri M, Kumar R, Ali I, Hussain SM, et al. Probabilistic approach to covid-19 data analysis and forecasting future outbreaks using a multi-layer perceptron neural network. *Diagnostics* (2022) 12:2539. doi:10.3390/diagnostics12102539

17. Mabrouki J, Azrour M, Dhiba D, Farhaoui Y, Hajjaji SE Iot-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. *Big Data Mining and Analytics* (2021) 4:25–32. doi:10.26599/BDMA.2020.9020018

18. Shah J, Mishra B Iot enabled environmental monitoring system for smart cities. In: 2016 International Conference on Internet of Things and Applications (IOTA); January, 2016; Pune, India (2016). p. 383–8. doi:10.1109/IOTA.2016.7562757

19. Mukherjee B, Wang S, Lu W, Neupane RL, Dunn D, Ren Y, et al. Flexible iot security middleware for end-to-end cloud–fog communication. *Future Generation Comp Syst* (2018) 87:688–703. doi:10.1016/j.future.2017.12.031

20. Rahut Y, Afreen R, Kamini D, Gnanamalar SS Smart weather monitoring and real time alert system using iot. *Int Res J Eng Tech* (2018) 5:848–54.

21. kumar AS, Murugan S, Elngar AA, Garg L, Kanmani R, Malar ACJ *A novel scheme for an IoT-based weather monitoring system using a wireless sensor network*. Cham: Springer International Publishing (2020). doi:10.1007/978-3-030-38516-3_10

22. Khan J, Li JP, Ahamad B, Parveen S, Ul Haq A, Khan GA, et al. Smsh: secure surveillance mechanism on smart healthcare iot system with probabilistic image encryption. *IEEE Access* (2020) 8:15747–67. doi:10.1109/ACCESS.2020.2966656

23. Khan J, Li JP, Haq AU, Khan GA, Ahmad S, Abdullah Alghamdi A, et al. Efficient secure surveillance on smart healthcare iot system through cosine-transform encryption. *J Intell Fuzzy Syst* (2021) 40:1417–42. doi:10.3233/jifs-201770

24. Ouni R, Saleem K Framework for sustainable wireless sensor network based environmental monitoring. *Sustainability* (2022) 14:8356. doi:10.3390/su14148356

25. Khan J, Khan GA, Li JP, AlAjmi MF, Haq AU, Khan S, et al. Secure smart healthcare monitoring in industrial internet of things (iiot) ecosystem with cosine function hybrid chaotic map encryption. *Scientific Programming* (2022) 2022:1–22. doi:10.1155/2022/8853448

26. Tripathy SK, Mondal SR, Nayak MR, Palai G Experimental studies on electronic smart device for automobiles application. *Opt Quan Elect* (2023) 55:550. doi:10.1007/s11082-023-04789-7

27. Li S, Ye J Design of rural ambient intelligence monitoring system based on ant colony optimization algorithm. In: 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE); November, 2023; Ballari, India (2023). p. 1–5. doi:10.1109/AIKIIE60097.2023.10390270

28. Lima J *Arduino vs Raspberry vs libelium vs WeIO vs rascal: what IoT board should you buy? Report*. London, United Kingdom: New Statesman Media Group Ltd (2022).

29. Sl LCD *Waspmote technical guide*. Aragon, Spain: Libelium (2020).

30. Atif M, Muralidharan S, Ko H, Yoo B Wi-esp—a tool for csi-based device-free wi-fi sensing (dfws). *J Comput Des Eng* (2020) 7:644–56. doi:10.1093/jcde/qwaa048

31. Libelium. *Encryption libraries. Report*. Aragon, Spain: Libelium (2017).

32. Stallings W *Cryptography and network security principles and practices*. London, United Kingdom: Pearson (2006).