Journal of Applied Engineering and Technological Science
Vol 6(1) 2024: 754-766

FLEET CONTROL WITH IOT USING TLS CERTIFICATES AND SIM7000G GPS DEVICE

René Cruz-Guerrero[1]*, José Epifanio Rojas-Cortés[2], German Cuaya-Simbro[3], Isaías Simón-Marmolejo[4]
Department of Computer Systems, National Technologic of Mexico, ITESA, México[123]
Department of Industrial Engineering, UAEH, México[4]
rcruz@itesa.edu.mx[1]*, 21030727m@itesa.edu.mx, gcuay@itesa.edu.mx[3],
isaiasm@uaeh.edu.mx[4]

Received: 15 July 2024, Revised: 05 November 2024, Accepted: 11 November 2024
**Corresponding Author*

ABSTRACT
Companies that apply technology to monitor mobile units face several challenges such as obtaining the best precision in geolocation coordinates, having good communication quality and security in data transmission, detecting route deviations, avoiding delays in their journeys, among others. This article proposes a real-time transport fleet monitoring system that uses the Internet of Things (IoT) to contribute to some of these challenges and improve the overall user experience of the collected data. A model is presented that includes a proposal to reinforce security in data transit with MQTT using Transport Layer Security (TLS) and a study carried out with the purpose of detecting which GPS coordinate acquisition device is more precise. 368 tests were carried out in five different coordinates with SIM800, ESP32, SIM7000G considering altitude, latitude, longitude, speed, number of satellites and sending time. The device that provided the best results in coordinate accuracy was the SIM7000G, even meeting the requirements for battery life and storage capacity due to the demand required to execute TLS security certificates.

Keywords: *Internet of Things, Security, GPS, Tracking, TLS, MQTT, SIM7000G.*

1. Introduction

The emergence of the Internet of Things (IoT) has presented promising solutions when used in monitoring or tracking tasks using the Global Positioning System (GPS). Its application is very broad and ranges from monitoring people for health purposes (Mahapatra et al., 2022; Rejeb et al., 2021), to tracking pets (Harper et al., 2023), to having systems that monitor fishing shipments (Tassetti et al., 2023), food traceability (Garaus & Treiblmaier, 2021), agriculture (Friha et al., 2021) among others. The monitoring and tracking of automotive units using GPS has been a rapidly growing field since the end of the 20th century, driven by the need to improve the efficiency, safety and control of vehicle fleets. This system allows the geographic location of vehicles to be tracked in real time using satellites, which has transformed transportation management at both a commercial and personal level.

Tracking transport units has become a critical need for many companies, especially those that depend on efficient and safe logistics. However, the lack of an adequate monitoring system can generate problems that affect both the operation and costs and safety of a company.

Companies that do not track their vehicles effectively often face problems of inefficiency in route management, increased travel times, higher fuel consumption, among others. These efficiency problems not only impact the quality of service, but also significantly increase operating costs.

The need to track transport units responds to a series of problems that directly affect efficiency, safety, operating costs, personnel control, customer satisfaction, and legal compliance. Therefore, the implementation of GPS tracking technology is presented as a key solution to address these challenges and improve the overall management of transport fleets (Monsreal & Carmona, 2021).

Regarding technological integration, many companies struggle to transition from prototype AI solutions to fully operational systems, in some cases requiring compliance with certain transport documentation (Jachimowski et al., 2022). These situations hinder the ability to track transport units effectively, the lack of machine learning and knowledge engineering skills within small and medium-sized businesses further complicates the integration of AI-based components into transport logistics (Jüngling et al., 2022).

Regarding forecasting issues, the dynamic nature of transport markets, especially in politically unstable regions, creates unpredictable demand patterns, making it difficult for companies to keep accurate track of transport units, due to this, companies must adapt their logistics strategies to cope with these fluctuations, which can strain existing tracking systems (Siryk & Siryk, 2021).

Currently, fleet control over mobile units faces various challenges to improve, such as the level of security of the data transported through communication control, the precision with which the geolocation coordinates of the units are obtained, the control of routes where communication is difficult, the implementation of artificial intelligence algorithms to control routes in real time, among others.

This paper proposes a model for companies that need to implement an IoT solution to monitor a fleet of trucks. It focuses mainly on suggesting the best solution regarding the device with which the acquisition of geolocation data is carried out in order to provide the highest precision of the coordinates. A security method was proposed to better protect the data transported by the MQTT communication protocol. Finally, regarding the architecture to be implemented, it was proposed to add a data integration module to the acquisition layer, which consists of incorporating necessary information provided by the company's ERP related to the indicators of mobile transport units, drivers and routes.

2. Literature Review

Fleet monitoring using IoT with GPS is a topic of interest in the field of Internet of Things (IoT) and transportation. To date, various studies have been developed on the monitoring and control of transport fleet.

Regarding works that challenge improving security in the implementation of IoT, in (Achary et al., 2023), in order to increase the protection of their data, they use the Constrained Application Protocol (CoAP) by adding elliptic curve cryptography (ECC) methods, which showed efficiency in smaller key sizes compared to traditional cryptographic methods such as RSA. In (Khalil et al., 2020), in order to strengthen the CoAP protocol and create another layer of protection for information, they used TACACS+ (Terminal Access Controller Access-Control System Plus) to perform authentication and authorization of devices in isolation, thereby protecting the main resources they use in the implementation of IoT, they configured the mobile phone as a client and the Raspberry as a server, demonstrating that their proposed method is compatible with the various commonly used devices. In other studies, security is implemented using the OSCORE (Object Security for Constrained RESTful Environments) standard, including the studies (Gunnarsson et al., 2021; Höglund et al., 2023).

Other works instead of using CoAP, apply the MQTT (Message Queue Telemetry Transport) protocol because it allows the publication-subscription transmission type, in (Stoev et al., 2020) they added a method called Payload encryption with which they protect the data on the client side, additionally they use a class to implement the Wi-Fi protocol so that it interacts with ESP8266 devices. For its part, in (Patel & Doshi, 2020) they present a security model to improve the services of the MQTT protocol in IoT systems, incorporating robust encryption, user authentication and data integrity measures to protect communication between IoT devices and intermediaries, they propose certificate-based authentication to address vulnerabilities such as interference and unauthorized access.

Considering the type of device for acquiring the geolocation coordinates of mobile units, in (Raikar et al., 2023) they track GPS transport units in real time using the NEO 6N sensor and the NodeMCU card to capture the data, they used Geofencing technology that helps them manage vehicle fleets, monitor their movements, optimize routes, track deliveries and provide real-time updates to customers by sending a notice when a vehicle enters or leaves a certain area.

Using a Sim908 with GSM and GPS modules in (Özdemir & TUĞRUL, 2019), they studied ways to improve the accuracy of real-time tracking systems, considering that environmental conditions can cause inaccuracies and delays in GPS signals. They applied

Kalman filter and logistic regression to minimize the error margin of the GPS data by an average of 15 meters to a considerably more accurate level. In (Tyagi & Sreenath, 2023), examines an Internet of Things system with GSM support for regulating the speed of electric cargo cars. Using wireless sensors, the system keeps an eye on a number of factors, such as tire pressure, motor temperature, vehicle speed, motor speed, truck payload, battery state of charge (SoC), battery state of health (SoH), and proximity to other vehicles.

In (Karne et al., 2022), to obtain data they use the Arduino UNO microcontroller in which they include a programming module together with the real-time clock (RTC) module, they developed an Android application that allows users to obtain information about mobile units based on their origin and destination, they use the MQTT protocol that allows them a distributed backend, with the grouping of intermediary devices the system becomes highly available, fault tolerant and scalable. The work of Desai and Phadke (2017) uses the Arduino ATMEGA card and a SIM808 GPS/GSM/GPRS module, their study focuses on vehicle tracking and monitoring to evaluate the main variables such as location, vehicle speed, engine temperature and fuel consumption. For its part, the collected data is displayed through a developed web interface and integrated with Google Maps to provide real-time tracking of the vehicle's location.

Bahr & Awad (2019) developed a system using GPS/GPRS/GSM and SIM900 sensors, communicated to a Raspberry Pi3, the acquired data can be sent to a mobile device application with Android operating system. In (Idris et al., 2024), they analyze the accuracy of real-time monitoring with the Blynk platform using information obtained from GPS modules including a simple and compact smart vehicle monitoring anti-theft system using the TTGO T-CALL ESP32 SIM800l. The data is analyzed regarding altitude and speed accuracy parameters. According to their results, they report that the TTGO T-Call ESP32 SIM800l has a lower percentage of latitude and longitude errors. In (Moumen et al., 2023) they present a real-time GPS tracking solution for connected vehicle networks using Arduino UNO R3, SIM8001, NEO6M GPS with V2X communication and VANET technologies.

Regarding the works that use the NEO6M GPS sensor on devices such as ESP32 or SIM800l, only (Moumen et al., 2023) and (Özdemir & TUĞRUL, 2019) try to improve the accuracy of the coordinates obtained by the device, obtaining better results in the second study, where Özdemir & TUĞRUL managed to reduce the error rate by combining the Kalman filter, logistic regression analysis and the moving average filter, achieving a decrease in the error margin of up to 15 meters. In the present work, the challenge of obtaining more accurate geolocation coordinates was addressed through a study where three of the most current devices on the market were evaluated with various tests to propose the one with the best performance, which will be explained in detail in the methodology section. Table 1 summarizes the main contributions of studies carried out that focus mainly on data processing, communication methods and acquisition devices.

Table 1 - Contributions of developed works.

Reference	Contribution
Archary et al., 2023	Integration of elliptic curve cryptography in CoAP protocol.
Khalil, 2020	They combine the CoAP protocol with TACACS+.
Stoev et al., 2020	Increase data security with Payload encryption
Özdemir & TUĞRUL, 2019	Minimize the margin of error of the GPS data with Kalman and regression
Tyagi & Sreenath, 2023	Store information in the cloud and perform complex data analysis
Raikar et al., 2023	It incorporates mobile applications and takes the geographical area in JSON format

In the studies where CoAP is used as a communication protocol, in (Archary et al., 2023) the increase in data security is done by applying encryption methods and in (Khalil et al., 2020) they do so by implementing the Terminal Access Controller Access Control System (TACACS+) where they focus on authenticating users trying to access network devices.

In the works that use MQTT, (Stoev et al., 2020) they implement security by adding a method called Payload encryption with which they protect the data on the client side and in (Patel & Doshi, 2020) they do it through an algorithm that strengthens MQTT with the integration of a Merkle tree, adding a layer of protection with the authentication and authorization plugins, thus reinforcing the security protocols and increasing resilience to potential threats. The proposal to improve security in our work consists of adding a layer for data protection through the use of TLS certificates in the MQTT Server broker.

3. Methodology

To develop a solution with IoT technologies that allows automating the control and monitoring of fleets, it is important to make an appropriate choice about the resources, platforms, equipmentand devices to use. Figure 1 shows the methodology that includes the necessary stages to obtain an adequate solution.

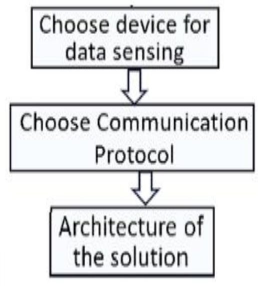

Choose device for data sensing
Choose Communication Protocol
Architecture of the solution

Fig. 1. Methodology

Before proposing the IoT architecture to be used, studies were carried out to choose the best resources to use in data acquisition and the communication protocol.

Regarding the sensor, the GPS module is a remote monitoring system that focuses primarily on environmental parameters in order to find the position of a device and receive data from satellites to know exactly where it is located. NEO-6 GPS devices are easily connected receivers, they have a UART, SPI, DDC (I2C) and USB communication interface. They are designed to be small in size, low in battery consumption, requiring an approximate current intensity of 37 mA. Comparing internally two of the most affordable sensors on the market NEO-6G with NEO-6M, although they have very similar characteristics, the 6M was chosen due to its minimum and maximum voltage range allowed (2.7 - 3.6) unlike the NEO-6G whose range is between 1.75 and 2 V.

3.1 Choose device for data sensing

To carry out the data acquisition, it was previously necessary to carry out a study to choose the best alternative. Three devices that use the NEO6M sensor (ESP32, SIM800 and SIM7000) were configured and a comparison was made to choose the one that most accurately provided the location coordinates.

ESP32 consists of a low-consumption microcontroller that integrates a Wifi and Bluetooth antenna that allows you to connect to different devices and networks, includes the NEO6M GPS receiver to obtain geographical location. SIM800 is composed of a LilyGo Sim800 module that includes cellular communication compatible with GSM and GPRS to transmit data, providing the device with internet connectivity, the NEO-6M GPS receiver and a battery of 10000 milli amps.

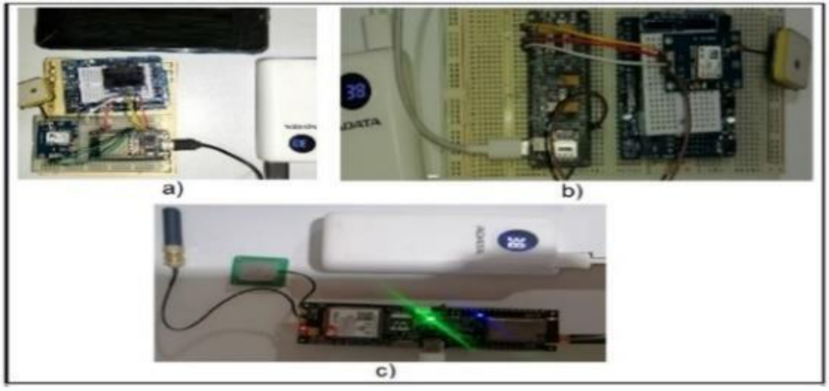


Fig. 2. Circuit of the three devices: a) ESP32 withcell phone b) SIM800 and c) SIM7000G.

Finaly, SIM 7000G is a device that includes cardwith Bluetooth, Wifi, telephone network, USB type C connection, power supply for solar panel, slot for 18650 battery, SIM card slot for

GS chip, configuring the GPS integrated into the card. Figure 2 shows the circuits created to perform tests and verify which of them presents the best performance.

In addition to obtaining the latitude, longitude and altitude with the three devices, the tests carried out also recorded the equivalent coordinates provided by Google Maps via mobile device, in order to compare and know the margin of error of the data provided by the devices with the NEO6M GPS sensor. To find the error distance, the Haversine formula was applied using the latitude and longitude obtained with the device and the Google Maps coordinate.

To obtain the distance between two coordinates o locations it is necessary to consider the earth's curvature. This situation is contemplated by the Haversine formula (Azdy & Darnis, 2020), whose purpose is to return the distance in meters between two positions, both specified as decimal degrees with latitude and longitude signs. Uses great circle distance calculation for a hypothetical sphere of radius 6,372.795 meters.

To use the Haversine formula, in addition to considering latitude and longitude, it is necessary to consider the radius of the Earth. This value is relative to the latitude, since the Earth is not perfectly round, the value of the equatorial radius is 6378 km while the polar radius is 6357 km. The Haversine formula for the spherical approximation of the distance (d) between two points on the Earth's surface is shown in Equation 1.

$$d = \mathit{2rarcsin}\,\left(\sqrt{\sin^2\left(\frac{\varphi2-\varphi1}{2}\right) + \cos(\varphi1)\cdot\cos(\varphi2)\,\sin^2\overset{\lambda}{\left(\frac{\lambda2-\lambda1}{2}\right)}}\right) \qquad \textbf{(1)}$$

where φ1, φ2, λ1 and λ2 refer to latitude and longitude, both expressed in radians to pass to trigonometric functions, r corresponds to the Earth's radius (Equatorial 6378.1 km, polar 6356.8 km, middle 6371.0 km). Where φ1 and φ2 correspond respectively to latitude y longitude obtained with Google Maps, while λ1 and λ2 y correspond respectively to latitude y longitude obtained with the device and GPS sensor.

To carry out the comparison between the three devices, a total of 368 tests were taken in different locations with a distance between them of at least one kilometer, recording data such as latitude, longitude, precision, accuracy, number of records per unit of time and quantity. of connected satellites. To carry out these tests, the modules were placed inside the mobile unit, verifying thatthey received signals from the satellites constantly. Each device was configured to report its position with a minimum of 4 satellites.

Table 1 shows the attributes of the dataset to be analyzed, which contains the class with three categories (names of the devices), the variables of the coordinates obtained with the devices themselves, the real coordinates provided by Google Maps, the error precision generated by the devices with respect to Google Maps calculated with the Haversine formula and additional attributes such as speed and number of satellites.

Because the objective is to determine which device provides the location coordinates with greater precision, the variables considered relevant for obtaining classification rules were Latitude_Dev, Longitude_Dev, Latitude_Maps, Longitude_Maps and Precision_error.

Table 1 - Attributes dataset.

Attribute	Description
Latitude_Dev	Latitude obtained by gps, a position distance north or south of the equator measured by degrees
Longitude_Dev	Longitude obtained by gps, position distanceeast or west of the prime meridian measured.
Altitude_Dev	Altitude obtained by gps, the distance between
Speed	the current position and the nearest point.
Longitude_Dev	A position's distance north or south of theequator
Satellites	measured by degrees.
Latitude_Maps	Longitude obtained by gps Number of satellites
Longitude_Maps	being trackedLatitude obtained by google maps
Precision_error	Longitude obtained by google mapsDevice
Delay_time Class	pressure error in meters
	Time elapsed between sending and receiving
	Classify dataset (ESP32, SIM_800, SIM_7000)

Machine Learning algorithms were applied to the integrated dataset to detect patterns that allow identifying correlations based on class. Methods that generate classification rules were applied to detect the characteristics of the attributes; it was possible to identify which device is more accurate. A CAR class association rule is a particular case of association rules (AR) that has the characteristic that it is composed of: an antecedent, formed by a set of elements or items and a consequent, which in the case of simple classification is a class (Ong et al., 2022). Two important elements that regularly describe a CAR and can determine if it is interesting are: support and trust. Support represents the frequency with which a CAR occurs in a set of transactions and confidence represents the probability that the consequent is present given the presence of the rule's antecedent.

Classification based in Association (CBA) is an algorithm that has two main components: a rule generator and a classifier constructor. Therefore, can be characterized as a two-stage algorithm since the rules and theclassifier are obtained at different times. The OneR algorithm is one of the simplest rule inductionmethods, selects the feature that carries the most information about the outcome of interest and creates decision rules from this feature (Frederika al., 2023). Ripple Down Rule learner (RIDOR) is also a direct classification method. RIDOR learns rules with exceptions by generating the default rule, using incremental reduced error pruning to find exceptions with the smallest error rate Ripper is stands for Repeated Incremental Pruning to Produce Error Reduction.

Table 2 - Rules Generated

Algorithm	Rules
CBA	1. Prec_error='(-inf-0.028]' 0 0 ==> Device=SIM7000 conf:(0.78), (73). 2. Prec_error = '(0.0465-inf)' ==> Device = esp32 conf:(0.69), (71). 3. Prec_error = '(0.0315-0.0465]' ==> Device = SIM800 conf:(0.68), (70).
OneR	Prec_error: < 0.0285 -> SIM7000 < 0.0385 -> SIM800 < 0.0395 -> esp32 >= 0.0595 -> esp32 (264/368 instances correct)
RIDOR	Device = SIM7000 (368.0/243.0) Except (Prec_error > 0.0285) and (Prec_error > 0.0445) => Device =esp32 (50.0/0.0) [31.0/0.0] Except (Prec_error > 0.0485) and (Prec_error <= 0.0595) and(Prec_error <= 0.0545) => Device = SIM800 (3.0/0.0) [1.0/0.0]
Ripper	Prec_error <= 0.031: SIM7000 (128.0/22.0) Prec_error > 0.046 AND Prec_error <= 0.048: esp32 (35.0/5.0) Prec_error > 0.042: SIM800 (47.0/21.0) Prec_error <= 0.039: esp32 (63.0/31.0) : SIM800 (15.0)
JRip	(Prec_error <= 0.04) and (Prec_error >= 0.04) => Device=SIM800(15.0/0.0) (Prec_error >= 0.038) and (Prec_error <= 0.038) => Device=SIM800(46.0/17.0) (Prec_error >= 0.049) and (Prec_error <= 0.059) => Device=SIM800(24.0/7.0) (Prec_error >= 0.032) => Device=esp32 (155.0/57.0) => Device=SIM7000 (128.0/22.0)

The best classification rules generated are shown in Table 2, as can be seen the most frequent and most important rules relate the variable Prec_error: < 0.0285 -> SIM7000 related to the SIM7000 device. Most of the detected patterns place the SIM7000 device as the one with the lowest precision error with an average range of less than .0315 kilometers (31 meters), in second place is the SIM800 with a margin between 0.0315 and 0.0465, and finally in last place is the esp32 device with an average range greater than 0.0465. This pattern is met in 81% of the instances of the analyzed dataset.

In the RIDOR method, the pattern obtained contains the SIM7000G class as an exception indicating that the precision error has lower values than the remaining classes where the Prec_error attribute is greater. According to the results obtained, the device that provided the best performance was the SIM7000G card, the TinyGPS++ GPS libraries were used for data acquisition, and the data sending protocol chosen was MQTT.

In addition, the SIM7000G device is designed for applications that require low-latency, low-throughput data communication in a variety of radio propagation conditions, can extend its battery life, and as shown in Figure 3, was designed to be able to work with solar charging.

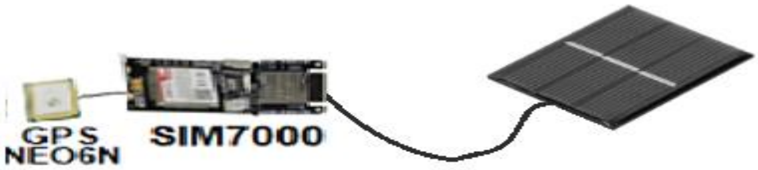


Fig. 3. SM7000G connected to solar cell

3.2 Choose communication protocol

For data transmission in IoT, the most appropriate sending and receiving protocol must be chosen depending on the type of data being handled. MQTT works with uninterrupted communication, as shown in Figure 4, it prevents the server from having to wait for requests from devices, constantly request new data, establish new connections, it only asks who needs the information and sends the messages (Mishra & Kertesz, 2020). It works through publications and subscriptions to a topic ("pub/sub" model), through a Broker (agent that manages publications and subscriptions). It is a lightweight protocol, ideal for IoT devices with limited resources, it is highly efficient in bandwidth and energy consumption, making it suitable for unstable networks.

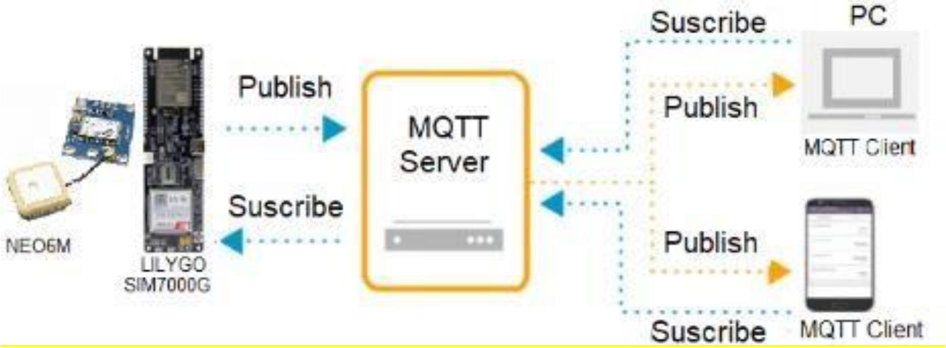


Fig. 4. Message Queue Telemetry Transport (MQTT) protocol.

On the other hand, the HTTP (Hyper Text Transfer Protocol) protocol works as a client-server (request-response) model, allowing interrelated communication between devices and transferring large amounts of data in small packets, which can cause large overheads, causing IoT air communication to cause some bandwidth problems.

Tests were performed with different time intervals on the number of messages received in both HTTP and MQTT, with different message sizes, the maximum being 56 MB. The results confirmed that the MQTT protocol allowed receiving a larger number of messages even in different time periods. Table 3 shows the results in summary.

Table 3 - Messages sent.

	Number of messages sent	
# Test	HTTPS	MQTT
1 min	628	2171
3 min	1928	5134
5 min	2858	9314

The fleet monitoring process ranges from collecting data from sensors to displaying the results ofthe indicators generated through data visualization techniques.

3.3 Architecture of the solution

The fleet monitoring process ranges from collecting data from sensors to displaying the results of the indicators generated through data visualization techniques. Considering the results shown in the previous sections, the SIM7000G device is suggested for the data acquisition stage and the use of the MQTT protocol for the transmission stage. Figure 5 shows the stages of the proposed IoT Architecture to carry out the entire process.

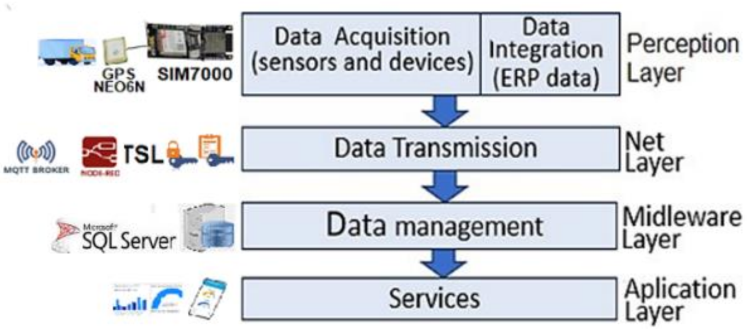


Fig. 5. IoT Stages of the proposed IoT Architecture

A three-layer IoT architecture only includes the acquisition, networking, and application stages, merging the data transmission and management layers into one. Due to this drawback, a four-layer architecture was chosen since having the data transmission or network stages and the data management stage separately makes it easier to address heterogeneity problems, and proposes a view that makes it easier to obtain, transport, present data, and manage the system (Burhan et al., 2018). In addition, this additional layer makes it easier to implement the recommended security mechanisms used to protect against intruders, allowing confirmation that the information is sent by authentic users and is protected from threats.

3.3.1 Perception Layer

This layer includes both data acquisition with the use of devices and data integration from corporate information.

Data Acquisition

With the selected device, the tasks corresponding to the acquisition layer were carried out, capturing the data emitted by the sensors and devices installed in the mobile units and their surroundings, emphasizing the monitoring of the values that were delivered by the GPS, mainly recording values of the number of satellites, latitude, longitude, altitude, among others. To achieve efficient communication between the different devices of the GPS systems, the MQTT broker was configured on the server, which acts as a receiver of the messages emitted by the connection modules that are on board the units involved. The version used was 5.0.20, as a complementary task it was also necessary to configure the EMQX broker service.

Data Integration

Unlike a typical IoT architecture, the first one, in addition to including a module to carry out the data acquisition process, also included a section with the objective of integrating necessary information that complements the results obtained with the sensors and devices. This is because the enterprise resource planning (ERP) system helps to provide data on some driver indicators such as travel times on each route, permitted routes and authorized places to refuel, among others. This information is very useful for the IoT system to report incidents, create triggers, issue alerts or generate alarms.

3.3.2 Data transmission

MQTT was chosen as the communication protocol, mainly due to its ability to send and receive a larger number of messages with client devices in real time, as well as its efficiency in the use of resources and the scalability it offers.

Its good performance was proven by sending messages of less than 256Mb to one or more devices with lower latency. Because large messages are not required for monitoring fleets of transport vehicles, this type of protocol works well, and this is even typical in IoT technology.

The MQTT broker allowed for handling a large number of simultaneous connections, including communication between different devices and another network. In addition, its ability to filter and route messages was verified, which, depending on the subscription topic, determines which client receives it.

Adding security with TLS

The combination of several devices in the use of IoT for the monitoring of transport fleets presents a series of security risks. The usual communication protocols in the Internet of Things lack data protection mechanisms, MQTT does not offer predetermined levels of security because it is designed to run on top of the TCP protocol. As an alternative solution, transport layer security can guarantee the security of MQTT by incorporating the TLS protocol.

Because TLS is a cryptographic protocol that uses an authentication mechanism to create a secure connection between the client and the server, it favors the implementation of encryption and authentication methods. Integrity protection for messages transmitted from the client to the server and vice versa can be achieved by using digital signatures as well as public and private keys. Figure 6, shows the proposed connection to add security over MQTT.

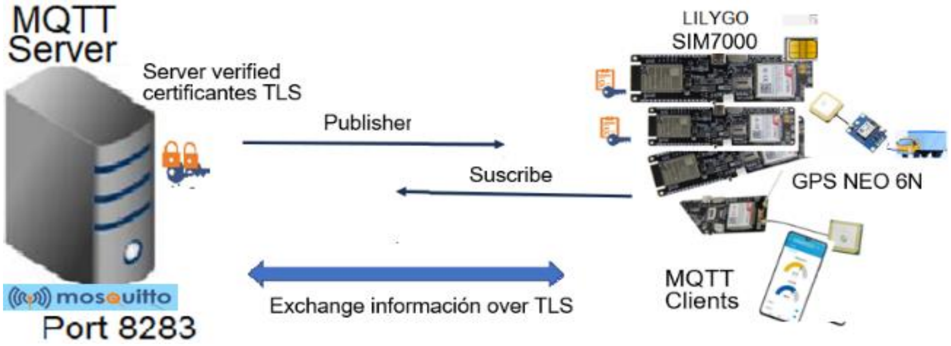


Fig. 6. Connection scheme to add security with TLS

MQTT messages transmitted as plain text over the (wireless) network are at risk of being read by some network traffic inspection software. To incorporate TSL, two main processes must be performed: creating certificates and keys for brokers and devices and configuring the MQTT server to use TSL. Figure 7 shows the suggested steps of the process to be performed to increase security by incorporating TSL.

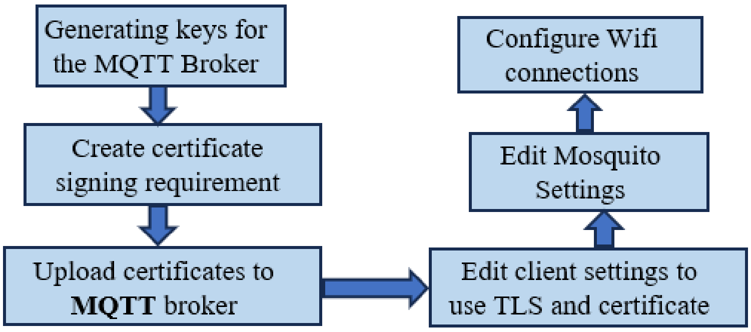

Generating keys for the MQTT Broker
Create certificate signing requirement
Upload certificates to MQTT broker
Edit client settings to use TLS and certificate
Edit Mosquito Settings
Configure Wifi connections

Fig. 7. Processes to add TLS to MQTT

First, in the MQTT broker, it is necessary to perform a complete configuration and generate all the necessary keys, then the necessary requirements are established to sign its certificates and finally all the required certificates are loaded.

Subsequently, on the client side, the necessary configurations are made so that the publication of MQTT messages can be carried out using TLS encryption, verifying that all clients have their corresponding certificate stored locally. Finally, the necessary configurations are made for the correct operation of the Wifi connections.

The certificates to be used correspond to the certificate of the authority and the mqtt server; for the mqtt-ca.crt authority certificate, the content of the open file is used as text and for the mqtt-srv server certificate, it is used in reduced fingerprint format. To use the WifiClient and PubsubClient components, the details of the certificates must be added. In the MQTT startup section, the parts are added so that the certificates are taken into account when making the Wifi connection.

The addition of TLS certificates over MQTT will require more storage and power resources, however, as far as clients are concerned, it is important to remember that MQTT has the advantage of being lightweight in its message transmission with minimal impact on power consumption. Therefore, as the SIM7000G device has the strengths of its 18650 lithium battery life (even with the alternative of connecting to solar panels) and its memory capacity, it makes it a good alternative to meet storage and power requirements.

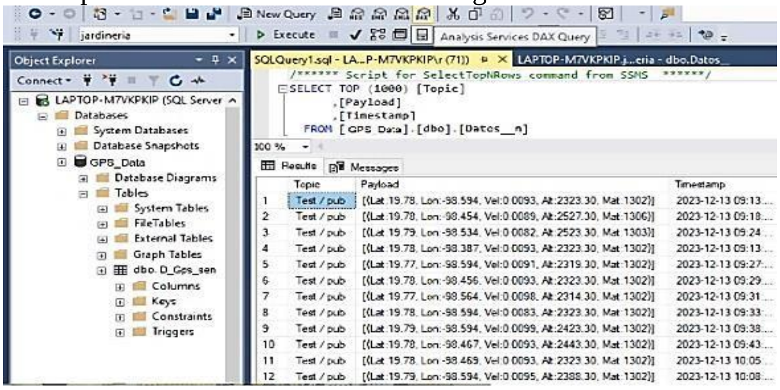
3.3.3 Data management

The database receives information related to the status of the vehicle, according to the control of the mechanism, compares the locations received by the units with the coordinates predefined by the organization according to its route list. At this stage it was necessary to add analysis and processing of the information collected before transferring it to the data center. To save the information, a database was created in the SQL Server handler and the following tasks were performed to connect to the Node-RED platform:

- Configuring SQL Server to accept remote connections.
- Adding the SQL Node in Node-RED to allow queries to be executed.
- Making the connection in the SQL node and configuring both the host and the port to connect to SQL Server.

Verifying the correct sending of data to the SMBD, configuring the flows in Node-RED to receive data from IoT devices and then store them in SQL Server using SQL queries.

The TinyGPS++ library was used to interpret and decode the data (attitude, longitude, speed, altitude, and exact time) received from the devices, which helped in the management of data acquired in real time. In conclusion, the device with the SIM7000G card, the MQTT protocol, and the TinyGPS++ GPS libraries were an efficient and effective combination to cover the communication and geolocation needs. After establishing the connection between the MQTT broker and the SMBD, the way in which the sensed information is stored in the database was verified; an example of these results is shown in Figure 8.



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Topic]
      ,[Payload]
      ,[Timestamp]
  FROM [GPS Data].[dbo].[Datos__n]
```

	Topic	Payload	Timestamp
1	Test / pub	[{Lat:19.78, Lon:-98.594, Vel:0.0093, Alt:2323.30, Mat:1302}]	2023-12-13 09:13:...
2	Test / pub	[{Lat:19.78, Lon:-98.454, Vel:0.0089, Alt:2527.30, Mat:1306}]	2023-12-13 09:18:...
3	Test / pub	[{Lat:19.79, Lon:-98.534, Vel:0.0082, Alt:2523.30, Mat:1303}]	2023-12-13 09:24...
4	Test / pub	[{Lat:19.78, Lon:-98.387, Vel:0.0093, Alt:2323.30, Mat:1302}]	2023-12-13 09:13:...
5	Test / pub	[{Lat:19.77, Lon:-98.594, Vel:0.0091, Alt:2319.30, Mat:1302}]	2023-12-13 09:27:...
6	Test / pub	[{Lat:19.78, Lon:-98.456, Vel:0.0093, Alt:2323.30, Mat:1302}]	2023-12-13 09:29:...
7	Test / pub	[{Lat:19.77, Lon:-98.564, Vel:0.0098, Alt:2314.30, Mat:1302}]	2023-12-13 09:31:...
8	Test / pub	[{Lat:19.78, Lon:-98.594, Vel:0.0083, Alt:2323.30, Mat:1302}]	2023-12-13 09:33:...
9	Test / pub	[{Lat:19.79, Lon:-98.594, Vel:0.0099, Alt:2423.30, Mat:1302}]	2023-12-13 09:38:...
10	Test / pub	[{Lat:19.78, Lon:-98.467, Vel:0.0093, Alt:2443.30, Mat:1302}]	2023-12-13 09:43:...
11	Test / pub	[{Lat:19.78, Lon:-98.469, Vel:0.0093, Alt:2323.30, Mat:1302}]	2023-12-13 10:05:...
12	Test / pub	[{Lat:19.79, Lon:-98.594, Vel:0.0095, Alt:2388.30, Mat:1302}]	2023-12-13 10:08:...

Fig. 8. Example of records incorporated into the database

3.3.4 Applications and services

This stage allowed to provide services to the end users of the IoT system such as the visualization module with dashboards and the system that interacts with the personnel involved. For this purpose, a web portal was developed, using specific technologies for both the frontend and the backend.

The development of the frontend for the web portal was programmed in Vue.js 3, a JavaScript framework with an intuitive interface, integrating various components in a modular way that includes advantages of reuse. It also allowed to integrate the modular and reusable components. On the other hand, PHP was used for the Backend of the web portal, showing high compatibility with the SQL Server manager. A module was also included that allows to show the results of the monitoring of the transport units in real time with the location coordinates, using a map; an example of a case is shown in Figure 9.

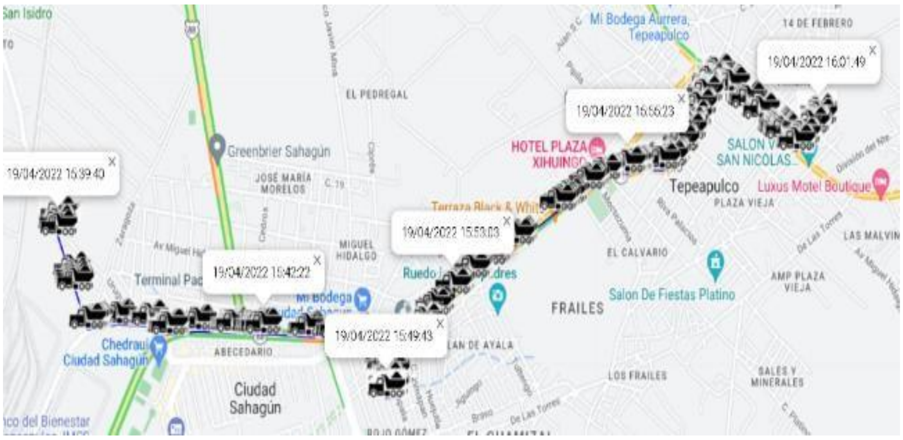


Fig. 9. Route reported by the GPS system

4. Conclusions

An IoT model was proposed for monitoring transport fleets in real time, where a strategy was applied to increase the security of the data that travels from the client to the MQTT broker and vice versa. To do this, the TLS cryptographic protocol was used because it works well with MQTT and handles light certificates.

Tests were carried out to find out which of the geolocation devices on the market provides the most accurate coordinates, using the values for altitude, latitude, longitude, speed, number of satellites obtained in each test. The performance of the SIM800, ESP32 and SIM7000G options was compared, where the latter provided the best results.

The results showed that the use of TLS improves the security of MQTT and prevents MITM threats from affecting the scanning of credentials and messages exchanged between the MQTT client and the broker. The combination of the developed device with the SIM7000G card with the MQTT protocol and the TinyGPS++ GPS libraries proved to be an efficient solution to cover the communication and geolocation needs. In conclusion, we can say that the use of TLS with MQTT is recommended to support security, especially when IoT devices have sufficient hardware resources. In the tests performed, SIM7000G proved to meet the battery life and storage capacity requirements for the execution of TLS security certificates.

References

Achary, R., Shelke, C. J., Marx, K., & Rajesh, A. (2023). Security Implementation on IoT using CoAP and Elliptical. Curve Cryptography *Procedia Computer Science, 230*, 493-502. https://doi.org/10.1016/j.procs.2023.12.105

Azdy, R. A., & Darnis, F. (2020, April). Use of haversine formula in finding distance between temporary shelter and waste end processing sites. In *Journal of Physics: Conference Series* (Vol. 1500, No. 1, p. 012104). IOP Publishing. https://doi.org/10.1088/1742-6596/1500/1/012104

Bahr, D. A., & Awad, O. A. (2019). LTE based vehicle tracking and anti-theft system using Raspberry PI Microcontroller. *Iraqi Journal of Information and Communication Technology*, *2*(1), 10-25.

Burhan, M., Rehman, R. A., Khan, B., & Kim, B. S. (2018). IoT elements, layered architectures and security issues: A comprehensive survey. *sensors*, *18*(9), 2796. https://doi.org/10.3390/s18092796

Desai, M., & Phadke, A. (2017, February). Internet of Things based vehicle monitoring system. In *2017 fourteenth international conference on wireless and optical communications networks (WOCN)* (pp. 1-3). IEEE. https://doi.org/10.1109/WOCN.2017.8065840

Frederika, A. A., Bayupati, I. P. A., & Buana, P. W. (2022). Classification Based Association (CBA) Menggunakan R. *Jurnal Ilmiah Teknologi dan Komputer, 3*(1), 1013-1019.

Friha, O., Ferrag, M. A., Shu, L., Maglaras, L., & Wang, X. (2021). Internet of things for the future of smart agriculture: A comprehensive survey of emerging technologies. *IEEE/CAA Journal of Automatica Sinica, 8*(4), 718-752.

https://doi.org/10.1109/JAS.2021.1003925

Garaus, M., & Treiblmaier, H. (2021). The influence of blockchain-based food traceability on retailer choice: The mediating role of trust. *Food control, 129*, 108082. https://doi.org/10.1016/j.foodcont.2021.108082

Gunnarsson, M., Brorsson, J., Palombini, F., Seitz, L., & Tiloca, M. (2021). Evaluating the performance of the OSCORE security protocol in constrained IoT environments. *Internet of Things, 13*, 100333. https://doi.org/10.1016/j.iot.2020.100333

Harper, S., Mehrnezhad, M., & Leach, M. (2022, November). Security and Privacy Concerns of Pet Tech Users. In *Proceedings of the 12th International Conference on the Internet of Things* (pp. 155-162).

Höglund, R., Tiloca, M., Bouget, S., & Raza, S. (2023, July). Key Update for the IoT Security Standard OSCORE. In *2023 IEEE International Conference on Cyber Security and Resilience (CSR)* (pp. 78-85). IEEE. https://doi.org/10.1109/CSR57506.2023.10225002

Idris, A., Iman, M. N., Sulong, S. M., Muhamad, W. N. W., & Bakar, A. A. (2025). Vehicle Tracking System Based on Internet of Things Utilizing TTGO T-CALL ESP32 SIM800l. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *45*(2), 118-127. https://doi.org/10.37934/araset.45.2.118127

Jachimowski, R., Zieliński, T., & Jóźwiak, A. (2022). The issue of transport documentation circulation between logistic processes participants. *WUT Journal of Transportation Engineering, 135*.

Jüngling, S., Fetai, I., Rogger, A., Morandi, D., & Peraic, M. (2022). On the track to application architectures in public transport service companies. *Applied Sciences, 12*(12), 6073. https://doi.org/10.3390/app12126073

Karne, R., Nithin, E., Sai Goutham, A & Rahul, A. (2022). An Internet of Things (IoT) enabled tracking system with scalability for monitoring public buses. *International Journal of Food and Nutritional Sciences, 11*. 9562-9574.

Khalil, K., Elgazzar, K., Abdelgawad, A., & Bayoumi, M. (2020, June). A security approach for CoAP-based internet of things resource discovery. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)* (pp. 1-6). IEEE. https://doi.org/10.1109/WF-IoT48130.2020.9221153

Mishra, B., & Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: A survey. *Ieee Access, 8*, 201071-201086. https://doi.org/10.1109/ACCESS.2020.3035849

Moumen, I., Rafalia, N., Abouchabaka, J., & Aoufi, M. (2023). Real-time GPS tracking system for IoT-Enabled connected vehicles. In *E3S Web of Conferences* (Vol. 412, p. 01095). EDP Sciences.

Mahapatra, S., Kannan, V., Seshadri, S., Ravi, V., & Sofana Reka, S. (2022). An IoT-Based Wristband for Automatic People Tracking, Contact Tracing and Geofencing for COVID-19. *Sensors, 22*(24), 9902. https://doi.org/10.1016/j.mehy.2019.109531

Monsreal, M. M., & Carmona-Benítez, R. B. (2022). Impact of IoT on supply chain performance. *Journal of applied research and technology, 20*(5), 584-593. https://doi.org/10.22201/icat.24486736e.2022.20.5.1957

Ong, H. F., Neoh, C. Y. M., Vijayaraj, V. K., & Low, Y. X. (2022, November). Information-Based Rule Ranking for Associative Classification. In *2022 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)* (pp. 1-4). IEEE. https://doi.org/10.1109/ISPACS57703.2022.10082812

Özdemir, Z., & TUĞRUL, B. (2019, October). Geofencing on the real-time GPS tracking system and improving GPS accuracy with moving average, Kalman filter and logistic regression analysis. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 1-6). IEEE. https://doi.org/10.1109/ISMSIT.2019.8932766

Patel, C., & Doshi, N. (2020). A novel MQTT security framework in generic IoT model. *Procedia Computer Science, 171*, 1399-1408. https://doi.org/10.1016/j.procs.2020.04.150

Raikar, M. M., Angadi, K., Reddy, S., Naik, N., & Doddwada, A. (2023, May). Fleet tracking and Geofencing using the Internet of Things (IoT). In *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)* (pp. 463-468).

IEEE. https://doi.org/10.1109/ICSCCC58608.2023.10176660

Rejeb, A., Treiblmaier, H., Rejeb, K., & Zailani, S. (2021). Blockchain research in healthcare: a bibliometric review and current research trends. *Journal of Data, Information and Management, 3*, 109-124. https://doi.org/10.1007/s42488-021-00046-2

Stoev, I., Zaharieva, S., Borodzhieva, A., & Staevska, G. (2020, July). An approach for securing MQTT protocol in ESP8266 WiFi module. In 2020 XI National Conference with International Participation (ELECTRONICA) (pp. 1-4). IEEE. https://doi.org/10.1109/ELECTRONICA50406.2020.9305164

Siryk, Z. O., & Siryk, O. Z. (2021). Managing a company's transport logistics: current challenges and development perspectives. *Regional economy*. https://doi.org/10.36818/1562-0905-2022-3-12

Tassetti, A. N., Galdelli, A., Pulcinella, J., Mancini, A., & Bolognini, L. (2022). Addressing gaps in small-scale fisheries: a low-cost tracking system. *Sensors, 22*(3), 839. https://doi.org/10.3390/s22030839

Tyagi, A. K., & Sreenath, N. (2022). Intelligent transportation system in internet of things-based computing environment. In *Intelligent Transportation Systems: Theory and Practice* (pp. 265-281). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-7622-3_12