IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151108

Smart Anti-Theft Smartphone Ecosystem with Offline Remote Access Using BLE and GSM

Bindu K.P[1], Dimple J[2], Gabburi Narasanna Pallavi[3], Gaddamadugu Dinavya[4]

Assistant Professor, Dept. of CSE, K.S. School of Engineering and Management, Bengaluru, India[1]

Student, Dept. of CSE, K.S. School of Engineering and Management, Bengaluru, India[2-4]

Abstract: Tracking apps that run on a phone stop working once a thief switches the handset off or cuts its connection, so the owner loses both the device and any way of reaching it. This paper presents an anti-theft ecosystem in which recovery is handled by a separate companion tracker with a battery of its own, a dedicated SIM card and a built-in satellite receiver. The tracker combines an ESP32-class controller, a GPS receiver and a SIM800L GSM/GPRS modem. It sends its position to an online dashboard, replies to location requests received as text messages, issues an SOS alert carrying a map link when a button is held, and accepts calls from the owner. A temporary Wi-Fi page stores the device identity, API key, access point name, upload interval and emergency numbers in flash memory, so no secret is written into the program. A Bluetooth Low Energy layer that detects separation from the phone is specified as the planned extension. The prototype was exercised through its configuration portal and its text-message interface; the observed behaviour, including a location reply carrying six satellites and an SOS alert delivered with a map link, is reported. Weaknesses of the present firmware and the next development steps are also discussed.

Keywords: Bluetooth Low Energy, GSM, GPS, anti-theft, offline tracking, SOS alert, ESP32, SIM800L, smartphone recovery.

I. INTRODUCTION

Smartphones now hold banking apps, authentication codes, work documents, photographs and private messages, so a stolen phone means lost money, identity and privacy and not only lost hardware. Tools such as Find My Device function only when the handset is switched on, logged in, connected and allowed to share its location. A thief typically breaks one of these conditions immediately, by powering the phone down, pulling out the SIM or turning on flight mode, and the tracking software inside the phone then has nothing to send.

A. Motivation and Challenges

The aim is to keep the owner in contact with a protected device when the phone itself is silent. Three properties are wanted: resilience, through a fallback channel that needs no IP connectivity; affordability, through commodity modules; and privacy, by sending little personal data and keeping credentials out of the code. The main difficulties are the small battery of a tracker, the need for a clear sky view, dependence on mobile coverage, and safe handling of keys and phone numbers.

B. Contributions

- An ecosystem concept that combines an independent GPS and GSM tracker with a planned BLE proximity layer.
- Integration and set-up of the tracker so that it uploads to a web dashboard, replies to text requests and raises SOS alerts.
- A Wi-Fi setup page that keeps credentials and numbers in flash, not in the source code.
- Test observations from the working prototype and a review of its present weaknesses.

II. PROBLEM STATEMENT

As phones increasingly hold sensitive personal, financial and professional data, theft of the device has become a serious worry. Recovery tools such as Find My Device depend heavily on an internet connection, so they stop being useful once the phone is turned off, loses connectivity or is tampered with. Continuous tracking and protection of the device are therefore not guaranteed. What is needed is a smart anti-theft system that keeps tracking the phone and alerting its owner in offline conditions, using technologies such as BLE and GSM.

A smart smartphone anti-theft system that enables tracking and alerts even when the device is offline using BLE, GSM, and GPS technologies.

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151108

III. Relevant Literature

A. Biometric IoT Anti-Theft System. Thamaraimanalan et al. secure a two-wheeler so that only an enrolled fingerprint can start the engine [1]. An ESP32 links a SIM800L for text alerts, a GPS module for live position, a relay that interrupts the fuel supply on tampering, and a camera that records evidence. The work shows that this controller, modem and GPS combination is suitable for alerting, yet it is built for vehicles and leaves all modules running, which increases idle power.

B. Portrait Recognition on STM32. Ye et al. build a system around an STM32 controller and a K230 edge unit that runs a light YOLO face detector, together with a PIR sensor, a servo pan-tilt mount and a GSM alert module [2]. Reported accuracy is 97.7%, and the PIR sensor reduces false alarms. Camera-based recognition can nevertheless be fooled, degrades with poor light or blocked views, and calls for a strong power source and a fixed site.

C. GSM and Arduino Automation. Tripathy et al. pair an Arduino board with a SIM900A modem that accepts text commands, validates them and drives relays [3]. The result demonstrates SMS as a dependable low-rate control path without internet access, although the setup is fixed in place and gives no position information.

D. BLE-Based Secure Tracking. Hernández-Goya et al. propose a framework with an Android application that works as a tracker on a phone flown on a drone and as a client on devices on the ground [4]. Location data travel inside BLE beacon frames secured with light-weight encryption. Without mobile coverage the drone must act as a relay to the server, and handling the keys adds further complexity.

E. BLE Key Locators. Dongre and Bamne review commercial BLE tags by range, price and parts, and argue that an ESP32 with BLE 5.0 is a good basis for building a tag [5]. These products depend on Bluetooth range and on the vendor's own ecosystem, so a tag is of no use after the item moves beyond radio reach.

F. Direction Estimation with LSTM. Cai et al. estimate Bluetooth angle of arrival with a compact neural network fed with raw IQ samples; it uses phase-difference features and fills gaps in angular training data by augmentation [6]. It outperforms baseline networks and generalises to unseen angles, but covers localisation only and needs an antenna array. It informs the BLE direction layer planned here.

IV. Comparative Analysis

Each reviewed system solves one part of the problem. The biometric and portrait systems [1], [2] give strong local verification but are tied to a vehicle or a fixed site and use considerable power. The GSM system [3] shows the value of text commands but has no position information. The BLE works [4]–[6] offer compact telemetry, ranging and direction estimation, yet are limited by short radio range, and [4] also needs aerial infrastructure. None combines an independent position source, a cellular fallback and a web dashboard in a portable unit for a smartphone, which is the combination targeted here.

V. Gap Analysis

- **No unified platform.** Local BLE trackers have no wide-area link once the item leaves radio range, while cellular systems have no sense of proximity to the phone. A single low-cost unit that joins the two is missing.
- **Dependence on the phone or the installation.** Phone-based tracking depends on the phone's power and connectivity, and camera and vehicle systems need a fixed installation. A portable companion with its own supply and SIM is needed.
- **Limited remote interaction.** Many trackers only broadcast or log positions. The owner also needs to query the unit by text, receive an alert on a button press and, if necessary, listen at the device.
- **Handling of credentials and location data.** Trackers often embed keys and numbers in their code or send data unprotected. Credentials should be entered at deployment time and stored in the device, and the data sent should be kept to a minimum.

VI. Proposed System Design

A. System Overview

Fig. 1 shows the ecosystem. The central element is an independent tracker unit that is powered separately and carries a SIM card and a GPS receiver of its own, so it does not rely on the smartphone's connectivity. The ESP32 controller reads positions from the GPS receiver and drives the SIM800L modem, over which it exchanges texts and calls with the owner and sends location records to an online dashboard. The ESP32 has a built-in BLE radio, so a proximity layer can be added later: a tag on the phone would let the tracker notice automatically when the two are separated. That layer is specified but has not been demonstrated, and the results in this paper come from the GPS and GSM part.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151108

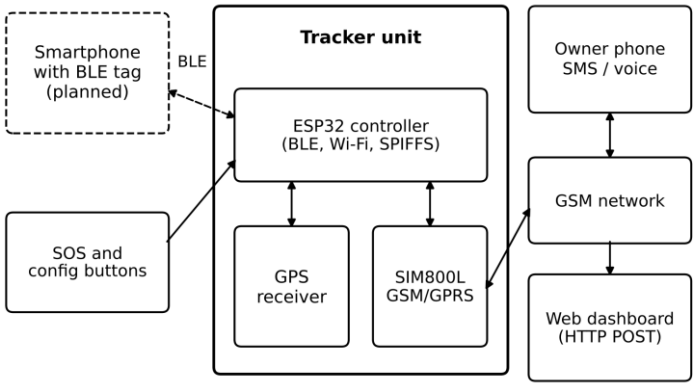


Fig. 1. Architecture of the ecosystem. The dashed parts are planned.

B. Workflow

- **Configuration:** after registering the device on the dashboard, the owner provides the device ID, key, APN, interval and up to three emergency numbers through the Wi-Fi portal.
- **Start-up:** the controller loads the saved settings, starts the GSM module, opens the GPRS bearer and sends a start-up text to the registered numbers.
- **Positioning:** the controller decodes GPS sentences and accepts a fix only with a valid position and at least four satellites.
- **Reporting:** at the chosen interval a JSON record with device ID, time and coordinates is posted to the dashboard.
- **Interaction:** a text reading SEND LOCATION from a stored number receives a reply holding a map link, the speed, the satellite count and the time; a long press of the SOS button sends the link to all registered numbers and calls the first one; incoming calls are answered automatically for up to 120 seconds.

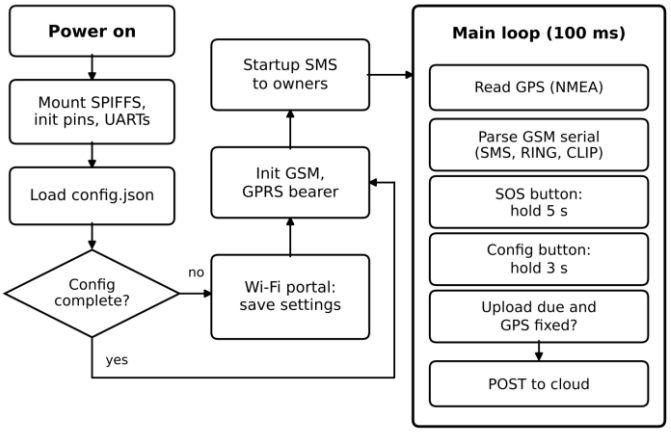


Fig. 2. Boot sequence and main loop of the tracker firmware.

C. Configuration and Persistence

At power-up the firmware mounts the SPIFFS flash file system and reads a JSON settings file. When the device ID, API key or APN is absent, or the configuration button is pressed during power-up, it starts a Wi-Fi access point that serves a one-page form with operator presets and a custom APN field, upload intervals from 15 seconds to 10 minutes, and fields for three emergency numbers. Entries with an empty key, ID or APN, or a primary number shorter than ten digits, are rejected. Accepted values are saved, Wi-Fi is switched off and the GSM start-up proceeds. Holding the configuration button for three seconds reopens the page.

D. GPS, GSM, and Message Handling

GPS sentences are decoded with the TinyGPS++ library [7]. For GSM, the firmware verifies that the SIM800L responds, enables text mode and caller identification, waits for network registration, sets the APN and opens the bearer, then checks that a non-zero IP address was assigned [8]. Incoming serial lines are classified as ring indications, caller-ID lines, call-ended notices or message headers. Sender numbers are compared on their last ten digits, so +91 and 0 prefixes do not cause mismatches. Messages from unknown senders are ignored.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151108

The main loop runs every 100 ms, and uploads are suspended while a call is active. A serial console offers diagnostic commands for status, forced upload, signal strength and volume.

VII. Results and Discussion

A. Test Set-up

The set-up consisted of the configured tracker holding its own SIM card with messaging and mobile data, the owner's smartphone to receive messages, a laptop running the serial monitor, and a location with an open sky for the GPS receiver. The functions were tried one at a time: reading stored settings, network registration, obtaining a GPS fix, uploading to the dashboard, answering a location text and sending the SOS alert. Only functions whose output could actually be seen were counted as working. Phone numbers and the key are hidden in every snapshot.

B. Configuration Portal

Fig. 3(a) and 3(b) show the portal opened by the tracker at its access-point address. The API key field is masked with a show toggle, a device ID is entered, an operator preset sets the APN, a 15-second upload interval is selected and three emergency numbers are entered before saving. After saving, the tracker went on to answer the text messages shown in Fig. 3(c).

C. Text-Message Interaction

Fig. 3(c) shows the owner's side of the conversation. The conversation opens with the device's help text, which the firmware sends in reply to any unrecognised message from a registered number. The first location request came back with a no-fix notice stating that zero satellites were visible, which shows that the firmware reports honestly instead of sending a stale or zero position. A later request returned a map link, speed 0.0 km/h and six satellites. An SOS alert was delivered as a text containing a map link. The two positions in the snapshot differ by about 4 m, which is consistent with a stationary receiver. No surveyed reference point was used, so this shows repeatability, not accuracy.

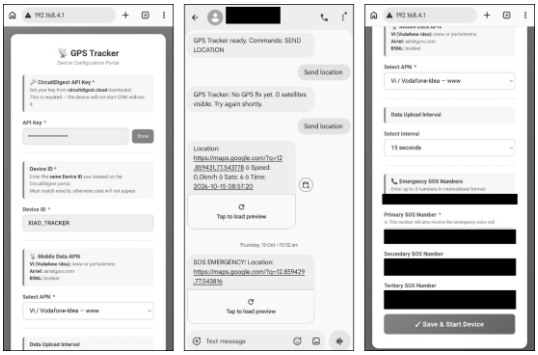


Fig. 3. Snapshots from the prototype: (a) and (b) configuration portal, (c) text-message interaction. Phone numbers are masked.

D. Discussion

The observations show that the tracker can be configured through a simple form and that the text interface behaves as designed, including the no-fix case, which is consistent with a receiver still acquiring satellites. These are single-trial observations on one unit, and quantities such as time to first fix, reply delay and current draw still need repeated measurements. Dashboard records and voice-call tests belong to the same test set and will be reported with them.

VIII. Conclusion and Future Work

We presented an anti-theft ecosystem whose recovery function sits in an independent tracker with a SIM and satellite receiver of its own, so it continues to work when the phone is off or offline. The prototype is configured through a Wi-Fi portal and, as the snapshots show, answers location texts and delivers SOS alerts; it is also set up to report positions to a dashboard.

Several steps are needed before real-world use. Commands are accepted on the sender's number alone, which can be forged, so a secret code should be added to each command. Calls are answered from anyone by default and should be limited to registered numbers. The upload uses plain HTTP and the API key is printed on the serial port, so encrypted transport and masked logs are advisable. Blocking delays in the main loop should be removed, and the 2G-only SIM800L should give way to an LTE module where 2G is withdrawn. The largest planned step is the BLE proximity layer, in which a tag on the phone lets the tracker detect separation automatically, with direction estimation using a model of the kind



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151108

described in [6], as outlined in our earlier survey [9]. A mobile application, geofencing and low-power modes are further extensions.

ACKNOWLEDGMENT

We sincerely thank our project guide and project coordinators for their valuable suggestions, guidance and support throughout the project. Their timely feedback on the design, implementation and testing of the tracker, and their constant encouragement, helped us overcome many difficulties and shaped this work into its present form.

REFERENCES

[1] T. Thamaraimanalan, S. Manmohan, R. Midhun, A. Raja Selvin and A. K. Ram Bahwan, "Design and development of a smart IoT-enabled biometric authentication-based intelligent anti-theft security system," in *Proc. 6th Int. Conf. IoT Based Control Networks and Intelligent Systems (ICICNIS)*, 2025, pp. 82–86.

[2] B. Ye, Q. Fan and J. Xu, "Design of a portrait recognition and intelligent tracking anti-theft system based on STM32 microcontroller," in *Proc. 7th Int. Conf. Frontier Technologies of Information and Computer (ICFTIC)*, 2025, pp. 575–579.

[3] J. Tripathy, S. Dash, R. Dash, J. Pal, S. Padhi and S. K. Mishra, "Designing a GSM and Arduino based reliable home automation system," in *Proc. 2024 OITS Int. Conf. Information Technology (OCIT)*, 2024, pp. 726–731.

[4] C. Hernández-Goya, R. Aguasca-Colomo and C. Caballero-Gil, "BLE-based secure tracking system proposal," *Wireless Networks*, vol. 30, pp. 5759–5770, 2024.

[5] S. Dongre and P. Bamne, "A survey on low energy Bluetooth based key locator and device detector components," *Int. J. Emerging Trends Eng. Res.*, vol. 11, no. 2, pp. 87–92, Feb. 2023.

[6] X. Cai, J. Yang, Y. Chen and R. Long, "Bluetooth AoA estimation method based on LSTM with spatial-structural feature fusion," *J. Syst. Eng. Electron.*, Feb. 2026.

[7] M. Hart, "TinyGPS++: A new GPS parsing library for Arduino," software library documentation.

[8] SIMCom Wireless Solutions, "SIM800 series AT command manual," manufacturer documentation.

[9] K. P. Bindu, J. Dimple, G. N. Pallavi and G. Dinavya, "Smart anti-theft smartphone ecosystem with offline remote access using BLE and GSM: A survey of hardware and algorithmic methods for securing devices and automated offline recovery," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 15, no. 5, May 2026.

 This work is licensed under a Creative Commons Attribution 4.0 International License