Journal of Science and Technology – Smart Systems and Devices
Volume 36, Issue 3, Month 2026, 000–000

A Navigation Message Authentication Approach to Mitigate GNSS Spoofing on Mobile Platforms

Ngoc Mai Xuan, Vinh La The, Hiep Hoang Van, Hung Pham Ngoc, Thuan Nguyen Dinh*
Hanoi University of Science and Technology, Ha Noi, Vietnam
**Corresponding author email: thuan.nguyendinh@hust.edu.vn*

Abstract
Global Navigation Satellite Systems (GNSS) provide the foundation for modern location-based services, yet civilian signals remain critically vulnerable to spoofing attacks that can compromise navigation integrity. While effective anti-spoofing countermeasures exist, they typically rely on specialized hardware such as multi-antenna arrays or software-defined radios, rendering them unsuitable for deployment on commodity smartphones. To bridge this gap, this paper proposes a practical, server-assisted Navigation Message Authentication (NMA) framework designed specifically for mobile platforms. Our approach introduces a lightweight verification layer where the mobile receiver cross-references locally decoded navigation data against cryptographically signed reference messages retrieved from a trusted server. The detection pipeline rigorously enforces both content consistency and timing sanity checks to identify and isolate varying classes of spoofing. Experimental validation, conducted using a custom Android application and the NAVISIM signal simulator in mixed-signal environments, demonstrates that the proposed method achieves a 0% False Negative Rate, effectively filtering out counterfeit signals in our test scenario. Although the exclusion of spoofed satellites inevitably results in reduced satellite availability and a consequent increase in the Root Mean Square Error (RMSE) due to geometric degradation, the system successfully preserves the integrity of the Position, Velocity, and Time (PVT) solution. This work contributes a deployable, software-defined defense mechanism that prioritizes authentic positioning over misleading precision, offering a potentially scalable software-based defense suitable for mobile devices. Similar to other NMA schemes (e.g., Galileo OSNMA), the proposed method remains vulnerable to replay/meaconing attacks when the delay between signal capture and retransmission is very short.

Keywords: GNSS, message authentication, mobile GNSS security.

1. Introduction

Global Navigation Satellite Systems (GNSS) including GPS, GLONASS, Galileo, and BeiDou are foundational to positioning, navigation, and timing (PNT) services in transportation, telecommunications, and safety critical mobile applications [1]. However, most civil GNSS signals provide limited built-in authenticity guarantees, and their low received power makes them vulnerable to spoofing attacks [2]. In a spoofing attack, an adversary transmits counterfeit GNSS-like signals that can be tracked by a receiver and ultimately drive it toward an incorrect PVT (Position, Velocity, Time) solution, potentially causing hazardous or financially costly outcomes.

Despite the ubiquity of GNSS on smartphones, practical spoofing mitigation on mobile platforms remains difficult. Mobile receivers typically lack a robust, deployable mechanism to verify whether decoded navigation data truly originates from satellites [3]. Meanwhile, many high-performing anti-spoofing techniques require specialized RF front-ends, multi-frontendsmulti-frontsmulti-antenna arrays, or computationally intensive signal processing that is hard to integrate into commodity phones under tight constraints on power, latency, and hardware access. As a result, current mobile devices can still accept counterfeit navigation messages as valid inputs to their PVT computation, undermining positioning integrity.

Prior work on GNSS spoofing mitigation can be broadly grouped into the following directions.

1.1. Signal Authentication

Cryptographic authentication attaches signatures or authentication codes to GNSS signals/messages to enable verification at the receiver [4–6]. In principle, this offers strong protection against generated spoofed signals, but it depends on constellation support, receiver capabilities, and timely key disclosure.

1.2. Detection Based on Signal Characteristics

These methods attempt to separate authentic and counterfeit signals by exploiting measurable differences in the received waveforms, such as abnormal signal power, correlation distortions, or other tracking-domain features [7, 8]. Direction-of-arrival (DoA) and spatial consistency checks can be particularly effective [9, 10]. However, such methods typically require multiple receivers or antenna arrays, which are not available on commodity smartphones.

p-ISSN 3093-3285
e-ISSN 3093-3315
https://doi.org/10.51316/jst.192.ssad.2026.36.3.x
Received: Jan 28, 2026; Revised: Apr 1, 2026;
Accepted: Apr 7, 2026; Online: Jun 1, 2026

Table 1. Comparison of GNSS Spoofing Detection Approaches

Approach Type	Suitable for Mobile	External Hardware Needed	Accuracy	Energy Efficiency
Signal characteristics	No	Yes (SDR)	High	Low
Direction-of-arrival	No	Yes (multi-antenna)	Very High	Moderate
Machine learning (DL)	Partially	No	High	Low
Proposed method	**Yes**	**No**	Moderate	**Moderate High**

1.3. Multi-antenna and Auxiliary-sensor Approaches

Multi-antenna systems leverage spatial processing (e.g., AoA/DoA estimation) to detect spoofing by identifying inconsistent arrival directions among signals carrying identical content [11–13]. Auxiliary information comparison (e.g., inertial sensors) can also increase robustness [14], yet these approaches may add integration complexity and energy overhead on mobile devices.

While the above techniques can be robust, their deployment on commodity mobile platforms is frequently constrained by the need for external hardware (e.g., SDR or antenna arrays), nontrivial calibration, or high computational cost. A high-level comparison is summarized in Table 1.

To improve authenticity for civil users, modern constellations have started introducing Navigation Message Authentication (NMA). For example, QZSS provides authentication services and can distribute authentication-related data via the L6 band [5, 15]. However, such capabilities are not yet broadly accessible on commercial smartphones, which commonly lack L6 reception and related processing support [16]. Therefore, there is a practical gap: mobile devices need an anti-spoofing mechanism that does not require specialized RF hardware or multi-antenna setups, and remains energy-efficient and deployable in real-world phone software stacks.

This paper presents a server-assisted method for verifying the authenticity of GNSS navigation messages on mobile devices. The key idea is to compare navigation data decoded (or reconstructed) on the phone against authenticated navigation messages obtained from a trusted server. By identifying mismatches between locally received navigation content and the trusted reference, the system can flag spoofed satellites and exclude them from PVT computation.

Recent Assisted Navigation Message Authentication (ANMA) studies, especially the smartphone proof-of-concept work by O'Driscoll et al. [17, 18], primarily focus on demonstrating the feasibility of server-assisted authentication on Android smartphones using Galileo OSNMA-related information and a trusted Internet-side reference stream. In essence, this ANMA direction addresses a key smartphone limitation: unreliable real-time decoding of complete authenticated navigation data under weak signal conditions and power-constrained operation. In addition, ANMA implementations mainly perform assisted verification by comparing selected unpredictable bits and relying on trusted navigation information provided by the server as the reference. Our work is complementary but differs in emphasis and scope. Rather than centering on an OSNMA feasibility demonstration, we develop a deployable mobile spoofing-mitigation pipeline that performs server-assisted integrity verification of locally decoded navigation messages and uses the verification outcome for satellite screening in PVT computation under mixed authentic/spoofed conditions. Different from unpredictable-bit comparison alone, our framework is designed toward a digital-signature-based navigation-message authentication model, in which valid signatures are generated using a private key held exclusively by the authorized managing entity; therefore, only that authority can produce authentic signatures for navigation data. In our experiments, we enable "Force full GNSS measurements" on Android to increase raw measurement continuity across constellations and reduce duty-cycling effects, which better supports message collection and timing consistency. While the current implementation validates a practical server-assisted integrity-checking workflow in realistic smartphone conditions, our broader objective is end-to-end authentication of complete navigation messages across multiple GNSS constellations, with applicability to legacy and civil signals. This architecture avoids hardware modifications and multi-antenna requirements, making it suitable for deployment on existing smartphones.

The main contributions of this work are: (i) a practical server-assisted architecture for navigation-message verification on smartphones; (ii) a message comparison and decision pipeline that integrates with the PVT computation to suppress spoofed measurements; and (iii) an implementation-oriented design that emphasizes mobile feasibility (no external RF hardware, moderate high energy efficiency) while providing meaningful spoofing discrimination.

2. Proposed Methodology

To mitigate GNSS spoofing attacks on mobile devices, this study proposes a hybrid architecture that combines on-device navigation message extraction with server-side authentication. The core idea is to verify the integrity of satellite navigation messages by cross-referencing them with trusted data provided by a secure server. Only messages verified as authentic are used in the final PVT computation, thereby enhancing the robustness of GNSS-based applications on mobile platforms.

2.1. System Architecture

The proposed system mitigates GNSS spoofing by verifying the authenticity of navigation messages through a server-assisted architecture. It consists of two main components:

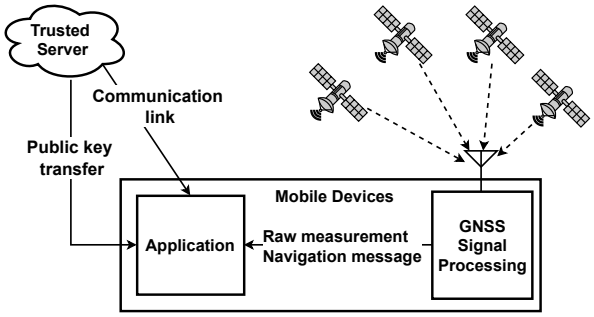

Fig. 1. System architecture

- Trusted Server: The server collects navigation messages from reliable sources, signs them using a public–private key pair, and stores the signed messages in a secure database. It exposes an HTTPS API for mobile clients to retrieve authenticated navigation data and signatures for verification.
- Mobile Devices: Smartphones receive GNSS signals from satellites and query the trusted server for corresponding signed reference messages. By verifying the digital signature and cross-checking the received navigation data against the trusted reference, the device identifies spoofed satellites and excludes them from the PVT computation.

Fig. 1 illustrates the system architecture and data flow between the trusted server and mobile devices.

2.2. Server-side

2.2.1. Server architecture

The trusted backend is deployed with a public–private network separation, as illustrated in Fig. 2. Mobile clients access the system over the public Internet through an API Gateway, which forwards requests to the Application API hosted in a private network. The Application API exposes two main interfaces: (i) a NAV Provider API for mobile devices to query signed navigation messages, and (ii) an Upload NAV API interface used to ingest trusted navigation data from

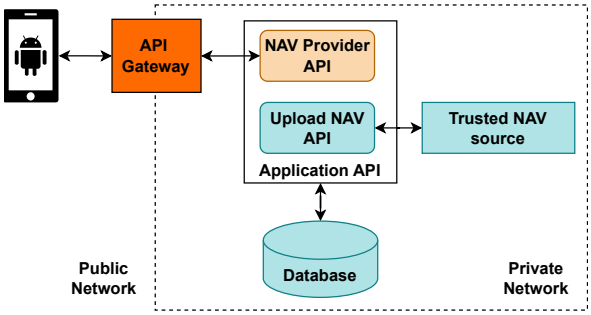

Fig. 2. Server-side architecture

trusted sources. All navigation messages, signatures, and related metadata are persisted in a database indexed for efficient lookup. Write operations (Upload NAV API) are protected by authorization policies, while client queries are served over HTTPS to ensure integrity in transit.

2.2.2. Trusted navigation message acquisition

To provide a high-integrity reference for mobile verification, we propose that the backend ingests navigation messages only from trusted sources, following a two-tier acquisition strategy. First, authenticated navigation data can be obtained from NMA-capable reference receivers leveraging authorized services, including Galileo OS-NMA [19], QZSS authentication services [20], and BeiDou authentication mechanisms [4]. These receivers serve as the primary trusted input to the server.

Second, to improve robustness in challenging environments, we propose deploying controlled monitoring stations (e.g., SDR-based receivers) in secure locations that perform high-confidence integrity screening prior to upload. Such stations can combine robust feature-based checks with computationally intensive deep learning spoofing detectors to filter suspicious signals before publishing navigation messages to the server [8]. In this design, only navigation messages that pass either NMA verification or station-level screening are accepted for signing, storage, and distribution to mobile clients.

2.2.3. Signing, storage, and server APIs

When a trusted NAV source submits navigation messages via the Upload NAV API interface, the server first canonicalizes the payload

$$m = \big(\texttt{gnssId}, \texttt{svId}, \texttt{week}, \texttt{tow}, \texttt{navMessage}\big), \quad (1)$$

to ensure a unique and consistent representation across different producers. The server then computes a cryptographic hash

$$h = H(m), \quad (2)$$

where $H(\cdot)$ is a secure hash function. Using the RSA private key sk, the server generates the digital signature

$$\sigma = \mathrm{Sign}_{sk}(h), \quad (3)$$

which serves as proof of integrity and server authenticity [21]. On the client side, the corresponding public key pk is used to verify the signature:

$$\text{Verify}_{pk}(h,\sigma) \in \{\text{true},\text{false}\}. \tag{4}$$

After signing, the server stores (m,σ) together with metadata (server timestamp, key identifier) in the database, indexed by `(gnssId, svId, week, tow)` for efficient lookup. For mobile verification, the NAV Provider API exposes an HTTPS endpoint that returns the signed navigation record. As illustrated in Fig. 2, the API is placed behind an API Gateway, which routes external requests to the private application network and enforces transport security.

The output format follows the structured JSON message in Listing 1.

Listing 1. Structure of Signed Navigation Message

```
{
  "gnssId": <int>,
  "svId": <int>,
  "week": <int>,
  "tow": <float>,
  "navMessage": "<string>",
  "signature": "<string>",
  "timestamp": "<string>"
}
```

2.3. Mobile Devices Side

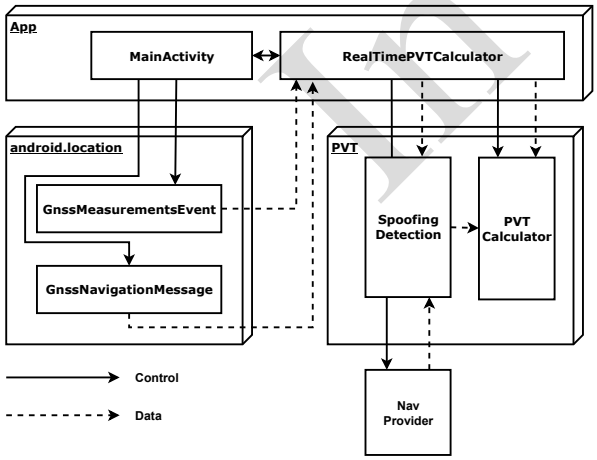

Fig. 3. Android application architecture

Fig. 3 presents the Android application architecture used for real-time spoofing detection and mitigation. The application is organized into three main modules. (i) *GNSS acquisition* collects raw measurements and navigation message subframes from the `android.location` GNSS stack. (ii) *Spoofing detection* performs server-assisted navigation message authentication by querying the trusted *Nav Provider* and applying the verification logic in Sections 2.2.3 and 2.3.3; satellites that fail verification are marked as untrusted and excluded. (iii) *PVT computation and visualization* computes position using only the retained satellites and updates the user interface in *MainActivity*. This modular separation enables integration of authentication logic at the application layer without modifying the underlying Android GNSS stack.

2.3.2. Navigation message collection on Android

The proposed approach relies on extracting navigation content directly on commodity smartphones. Starting from Android 7.0 (API level 24), the `android.location` framework exposes raw GNSS measurements and navigation message data through callbacks such as `GnssMeasurementsEvent` and `GnssNavigationMessage` [22, 23]. In our implementation, `GnssNavigationMessage` provides the decoded navigation message subframes (used as $\texttt{navMessage}_{rx}$), while `GnssMeasurementsEvent` supplies timing-related metadata (e.g., receiver clock information) required for aligning $(week, tow)$ and performing the timing sanity check described in Section 2.3.3. These on-device observations form the inputs to the spoofing detection module in Fig. 3.

2.3.3. Mobile device operations

On the client side, the smartphone performs navigation-message authentication per satellite and excludes untrusted satellites from the subsequent PVT computation. After decoding a local navigation message and obtaining its metadata $(gnssId, svId, week, tow)$ from the receiver, the device queries the *NAV Provider API* to fetch the corresponding signed reference record (Listing 1) from the trusted server.

Given the returned server record, the device reconstructs the canonical message m as in (1) and computes its hash $h = H(m)$ as in (2). Signature validity is checked using the server public key pk:

$$v = \text{Verify}_{pk}(h,\sigma), \tag{5}$$

where σ is the signature provided by the server. If $v =$ false, the record is rejected and the satellite is excluded.

If the signature is valid, the device performs a content-consistency test between the locally decoded navigation message $\texttt{navMessage}_{rx}$ and the server-provided message $\texttt{navMessage}_{srv}$:

$$c = \left(\texttt{navMessage}_{rx} = \texttt{navMessage}_{srv}\right). \tag{6}$$

A mismatch ($c =$ false) indicates potential spoofing or severe demodulation corruption and triggers exclusion.

Finally, the device applies a timing sanity check to raise suspicion of replay/meaconing by comparing the local reception timestamp t_{rx} and the server-side reception timestamp t_{srv}:

$$\Delta t = |t_{rx} - t_{srv}| \le \Delta t_{\max}, \tag{7}$$

where $\Delta t_{\max}$ is a predefined tolerance selected as an empirical network-latency tolerance rather than a pure RF propagation bound. In this networked smartphone–server setting, the observed timestamp difference is dominated by Internet access latency/jitter (e.g., 4G/Wi-Fi), OS scheduling effects, and clock offset between the smartphone and the trusted server-side receiver, while the geometric GNSS signal propagation difference between receivers contributes only a relatively small component (typically on the order of milliseconds).

Accordingly, the threshold is used as a practical anomaly-screening parameter: values within the expected operating range (e.g., a lower-bound region near nominal GNSS reception timing and an empirically calibrated upper bound around 150 ms) are treated as plausible, whereas larger discrepancies increase suspicion of delay-inducing attacks, including replay/meaconing or man-in-the-middle behavior. This check should therefore be interpreted as a heuristic consistency test, not a complete defense against low-latency replay/meaconing; a sufficiently capable adversary with very small added delay may still evade detection [24].

Satellites that satisfy (5–7) are retained for PVT computation; otherwise, they are excluded.

3. Experiments

To evaluate the proposed GNSS spoofing detection system, we conducted a series of controlled experiments conducted using both simulated and mixed-signal environments. The evaluation focused on the system's ability to accurately detect spoofed signals and maintain reliable PVT computation under spoofing conditions.

3.1. Experiments Setup

All experiments were conducted using the NAVISIM GNSS signal simulator developed by the Navigation Technology Research and Development Cooperation Center (NAVIS), Hanoi University of Science and Technology [25]. In this study, NAVISIM was configured to generate and transmit *GPS-only* simulated spoofing signals for selected GPS satellite IDs (PRNs). For navigation message generation, NAVISIM retrieves broadcast ephemeris data (GPS navigation messages) from the NASA Earthdata/CDDIS broadcast ephemeris product and synthesizes ephemeris-consistent GPS baseband signals prior to RF transmission [26].

The over-the-air test configuration is shown in Fig. 4. During testing, the smartphone receiver operates in a real outdoor environment and tracks signals from multiple GNSS constellations supported by the device. The test handset was a Xiaomi Redmi Note 12T Pro, running Android 13 on a MediaTek Dimensity 8200 Ultra platform, with GNSS positioning support for GPS (L1), GLONASS (G1), BeiDou (B1I/B1c), Galileo (E1), and QZSS (L1). The spoofing source, however, injects *GPS-only* simulated signals, so the attack is evaluated as a partial-constellation spoofing scenario in which forged signals are introduced only on selected GPS satellites while other live GNSS signals remain authentic.

In addition, the test smartphone was configured with the "Force full GNSS measurements" developer option enabled. When activated, this mode forces the Android GNSS stack to continuously track all visible constellations and frequencies without applying power-saving duty-cycling. As a result, the receiver outputs uninterrupted raw measurements at the highest available rate, ensuring that spoofing detection and mitigation algorithms operate on a complete, non-throttled GNSS dataset.

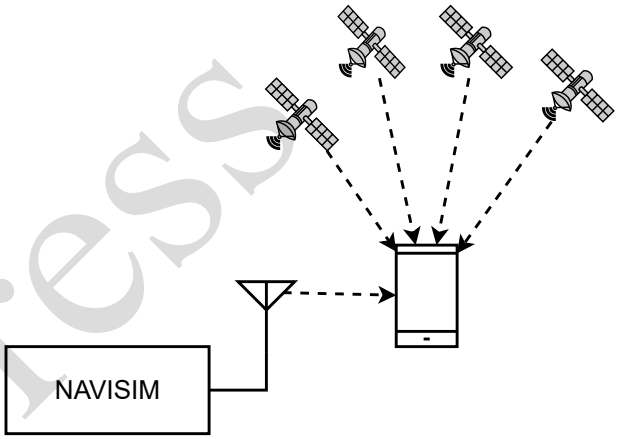

Fig. 4. Test environment setup

The smartphone runs a GNSS spoofing detection and mitigation application that identifies satellites carrying forged navigation data and excludes them from the PVT (Position, Velocity, Time) computation. Performance is assessed by the system's ability to (i) detect spoofed GPS signals and (ii) maintain positioning integrity after excluding identified spoofed satellites from the solution.

We designed two experimental environments:

3.1.1. Simulation-only environment

This environment evaluates spoof rejection when only simulated signals are present. NAVISIM transmits exclusively spoofed GPS signals while authentic satellite signals are absent, as illustrated in Fig. 5.

3.1.2. Mixed authentic and simulated signal environment

This environment evaluates performance under realistic reception conditions where authentic GNSS signals coexist with locally transmitted spoofed signals. The experiment was conducted on the rooftop of Building B1 at Hanoi University of Science and Technology, located at geographic coordinates (21.00460082, 105.84690533). The smartphone receiver was placed at the measurement point to track live

Fig. 5. Setup for the simulation-only environment with spoofed signals.

satellite signals from the open sky, while the NAVISIM signal generator transmitted simulated GPS signals from a nearby position. The separation between the smartphone and the spoofing transmitter was approximately 7 m, with a transmit power level of about 50 dB. Under this configuration, the receiver simultaneously observes authentic and simulated signals, requiring the system to preserve authentic satellites for PVT computation while correctly identifying and rejecting the injected spoofed signals.

Fig. 6. Setup for the mixed environment with authentic and spoofed signals.

3.2. Evaluation Metrics

We evaluate spoofing detection using a confusion matrix [27] (Table 2), which summarizes four outcomes: TP, FP, FN, and TN. From this matrix, we report the False Negative Rate (FNR) and False Positive Rate (FPR), capturing missed detections and false alarms, respectively:

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP}+\mathrm{FN}}, \qquad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}. \tag{8}$$

Table 2. Confusion Matrix for GNSS Spoofing Detection

	True	
Prediction	**Spoofing**	**Normal**
Spoofing	True Positive (TP)	False Positive (FP)
Normal	False Negative (FN)	True Negative (TN)

Positioning performance is evaluated by computing the two-dimensional root-mean-square error (RMSE) in meters, where each geodetic position fix (ϕ_i, λ_i) is first converted into its great-circle distance from a reference location $(\hat{\phi}, \hat{\lambda})$ using the Haversine formula [28]. For a spherical Earth model with radius $R = 6371000\,\mathrm{m}$, the great-circle distance d_i between the i-th sample and the reference point is

$$\Delta\phi_i = (\phi_i - \hat{\phi})\frac{\pi}{180}, \qquad \Delta\lambda_i = (\lambda_i - \hat{\lambda})\frac{\pi}{180}, \tag{9}$$

$$a_i = \sin^2\!\left(\frac{\Delta\phi_i}{2}\right) + \cos\!\left(\phi_i \frac{\pi}{180}\right)\cos\!\left(\hat{\phi}\,\frac{\pi}{180}\right)\sin^2\!\left(\frac{\Delta\lambda_i}{2}\right), \tag{10}$$

$$c_i = 2\arctan 2\!\left(\sqrt{a_i}, \sqrt{1-a_i}\right), \qquad d_i = R\,c_i. \tag{11}$$

The two-dimensional positioning RMSE is then defined as

$$\mathrm{RMSE}_{2D} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2}, \tag{12}$$

which provides an error metric expressed directly in meters and accounts for Earth curvature, thereby avoiding the incorrect use of Euclidean distance in latitude/longitude degrees.

Finally, we report Satellite Availability (SA), defined as the number of satellites retained for PVT computation after filtering (a valid solution typically requires at least four satellites). These metrics jointly characterize detection reliability and positioning robustness under spoofing.

4. Results

This section reports the reliability of the proposed spoofing detection pipeline under three conditions: (i) spoofed-only, (ii) mixed authentic–spoofed, and (iii) normal (non-spoofed).

Table 3. Comparison of Results Between Environments

Metric	Spoofed-Only	Mixed-Signal	Normal
FNR	0%	0%	–
FPR	0%	2.25%	1.21%
SA	0	29	55
RMSE_{2D}	Position fix failed	12.445m	3.97 m

Table 3 shows that, under the same spatial test condition, positioning accuracy is clearly lower when spoofing transmission is present. In normal operation, the system reaches $RMSE_{2D} = 3.97$ m, while in mixed authentic–spoofed reception, $RMSE_{2D}$ increases to 12.445 m (about 3.14× higher). At the same time, satellite availability drops from 55 to 29 after screening. This degradation is expected because the spoofing transmitter injects a strong GNSS-like RF component that perturbs receiver tracking and worsens post-filter geometry, so accuracy decreases even though integrity is protected.

For spoofing-detection performance, the method authenticates navigation messages by checking consistency with trusted server-side references. In the tested normal spoofing model (i.e., message-forging spoofing without replay/meaconing), the method achieves $FNR = 0\%$, meaning no spoofed satellite is accepted for PVT. Enabling "Force full GNSS measurements" improves raw-measurement continuity and decoding stability, which mainly helps reduce false alarms (lower FPR). However, in weak-signal conditions (e.g., underpasses) or during spoofing RF transmission, bit decoding can still become unstable and interact with network delay, increasing conservative mismatches; this explains why FPR rises from 1.21% (normal) to 2.25% (mixed-signal). In addition, because this is a server-assisted authentication scheme, practical operation requires a stable Internet connection; high latency, packet loss, or intermittent connectivity can delay trusted-message retrieval and further degrade real-time detection consistency. Overall, the operating point prioritizes spoof rejection reliability over accuracy/availability in challenging channels.

5. Conclusion and Future Work

This paper proposed a server-assisted navigation message verification approach to mitigate GNSS spoofing on mobile platforms. By cross-checking navigation messages decoded on the smartphone against trusted, authenticated reference messages, the system reliably identifies spoofed satellites and excludes them from PVT computation. Experimental results in the NAVISIM GPS-only spoofing setup indicate strong spoof rejection, while the Android smartphone receiver continues to track measurements from multiple live GNSS constellations. After excluding the identified spoofed GPS satellites, the receiver can still maintain a usable position solution, thereby preserving positioning integrity for safety and security-critical applications.

Importantly, the proposed architecture may increase the difficulty for attackers, since generating ephemeris-consistent forged messages alone may not be sufficient to bypass verification. However, this protection is not absolute: if an attacker can perform a replay/meaconing attack with sufficiently low latency in real time, the method may fail to defend against such attacks [24]. In addition, enabling "Force full GNSS measurements" can increase power consumption compared with Android's default GNSS operation (e.g., duty-cycled behavior), so this mode should be enabled with careful consideration of the energy–security trade-off

Future work will focus on four directions: (i) integrating secure time synchronization with Network Time Security (NTS) to provide authenticated time references and tighter delay bounds for more robust replay/meaconing detection; (ii) developing secure, scalable, and highly available server architectures, including authenticated APIs, hardened inter-service communication, and distributed verification services, and evaluating their latency, trust, and reliability trade-offs; (iii) improving mobile efficiency and detection robustness through elliptic-curve signatures such as ECDSA to reduce cryptographic overhead; and (iv) adaptive decision logic, and fusion of complementary smartphone indicators such as C/N_0, AGC/ADC-related features (when available), and learning-based classifiers on raw measurements.

Acknowledgment

This work was financially supported by the Ministry of Education and Training of Vietnam (MOET) under Project No. CT2025.EA.BKA.11. The authors sincerely acknowledge the support provided through this project.

References

[1] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, GNSS Position Integrity in Urban Environments: A Review of Literature, IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 9, pp. 2762–2778, 2018. https://doi.org/10.1109/TITS.2017.2766768.

[2] J. Zidan, E. I. Adegoke, E. Kampert, S. A. Birrell, C. R. Ford, and M. D. Higgins, GNSS Vulnerabilities and Existing Solutions: A Review of the Literature, IEEE Access, vol. 9, pp. 153960–153976, 2021. https://doi.org/10.1109/ACCESS.2020.2973759.

[3] L. Meng, L. Yang, W. Yang, and L. Zhang, A survey of GNSS spoofing and anti-spoofing technology, Remote sensing, vol. 14, no. 19, pp. 4826, 2022.

[4] X. Chen, R. Luo, T. Liu, H. Yuan, and H. Wu, Satellite Navigation Signal Authentication in GNSS: A Survey on Technology Evolution, Status, and Perspective for BDS, Remote Sensing, vol. 15, no. 5, 2023. https://doi.org/10.3390/rs15051462.

[5] K. Chino, D. Manandhar, and R. Shibasaki, Authentication technology using QZSS, in 2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014, 2014, pp. 367–372.

[6] Z. Wu, Y. Zhang, and R. Liu, BD-II NMASSI: An Scheme of Anti-Spoofing and Open BeiDou II D2 Navigation Message Authentication, IEEE Access, vol. 8, pp. 23759–23775, 2020. https://doi.org/10.1109/ACCESS.2020.2970203.

[7] E. Ghizzo, M. Hussong, J. Lesouple, C. Milner, A. Garcia-Pena, and C. Macabiau, Assessing spoofer impact on gnss receivers: Tracking loops, NAVIGATION: Journal of the Institute of Navigation, vol. 72, no. 4, 2025. https://doi.org/10.33012/navi.724.

[8] P. Borhani-Darian, H. Li, P. Wu, and P. Closas, Detecting GNSS spoofing using deep learning, EURASIP Journal on Advances in Signal Processing, vol. 2024, no. 1, pp. 14, 2024. https://doi.org/10.1186/s13634-023-01103-1.

[9] H. V. Hoang, H. Q. Pham, H. V. Nguyen, T. D. Nguyen, V. The La, T. H. Ta, and H. P. Ngoc, Enhancing GNSS spoofing detection by carrier-smoothing pseudorange double difference measurements, GPS Solutions, vol. 29, no. 4, pp. 189, 2025.

[10] E. Falletti, G. Falco, V. H. Nguyen, and M. Nicola, Performance Analysis of the Dispersion of Double Differences Algorithm to Detect Single-Source GNSS Spoofing, IEEE Transactions on Aerospace and Electronic Systems, vol. 57, no. 5, pp. 2674–2688, 2021. https://doi.org/10.1109/TAES.2021.3061822.

[11] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina, Receiver-Autonomous Spoofing Detection: Experimental Results of a Multi-Antenna Receiver Defense against a Portable Civil GPS Spoofer, in Proceedings of the 2009 International Technical Meeting of The Institute of Navigation, Anaheim, CA, Jan. 2009, pp. 124–130.

[12] A. Broumandan, A. Jafarnia-Jahromi, V. Dehghanian, J. Nielsen, and G. Lachapelle, GNSS spoofing detection in handheld receivers based on signal spatial correlation, pp. 479–487, 2012. https://doi.org/10.1109/PLANS.2012.6236917.

[13] J. Magiera, A Multi-Antenna Scheme for Early Detection and Mitigation of Intermediate GNSS Spoofing, Sensors, vol. 19, no. 10, 2019. https://doi.org/10.3390/s19102411.

[14] L. Junzhi, L. Wanqing, F. Qixiang, and L. Beidian, Research Progress of GNSS Spoofing and Spoofing Detection Technology, pp. 1360–1369, 2019. https://doi.org/10.1109/ICCT46805.2019.8947107.

[15] G. o. J. Cabinet Office, Quasi-Zenith Satellite System Interface Specification: Signal Authentication Service (IS-QZSS-SAS-001), 2023. [Online] URL: https://qzss.go.jp/en/technical/download/pdf/ps-is-qzss/is-qzss-sas-001_draft-002.pdf?t=1675691650065.

[16] Quasi-Zenith Satellite System, List of Products that Support QZSS, June 2024. [Online] URL: https://qzss.go.jp/en/usage/products/list.html.

[17] C. O'Driscoll, J. Winkel, and I. Hernandez, Assisted NMA proof of concept on Android smartphones. In 2023 IEEE/ION Position, Location and Navigation Symposium (PLANS), IEEE, 2023. [Online].

[18] C. O'Driscoll, I. Fernandez-Hernandez, J. Winkel, T. Willems, and A. Galan-Figueras, Assisted NMA with Secure Time Synchronization on Android Smartphones, in Proceedings of the 38th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2025), 2025, pp. 1132–1143.

[19] T. Hammarberg, J. M. V. García, J. N. Alanko, and M. Z. H. Bhuiyan, An Experimental Performance Assessment of Galileo OSNMA, Sensors, vol. 24, no. 2, 2024. https://doi.org/10.3390/s24020404.

[20] J. Quasi-Zenith Satellite System (QZSS) Operations Office, QZNMA: Start of signal authentication service, 2024. [Online] URL: https://qzss.go.jp/en/overview/notices/qznma_240401.html.

[21] Q.-L. Xu and T.-S. Chen, An efficient threshold RSA digital signature scheme, Applied Mathematics and Computation, vol. 166, no. 1, pp. 25–34, 2005. https://doi.org/10.1016/j.amc.2004.04.054.

[22] Google, android.location: Android Developers Reference, 2024. [Online] URL: https://developer.android.com/reference/android/location/package-summary.

[23] Google, Raw GNSS Measurements, Dec. 2024. [Online] URL: https://developer.android.com/.

[24] H. Wang, Y. Zhang, X. Zhu, J. He, S. Zhao, Y. Shen, and X. Jiang, Practical Spoofing Attacks on Galileo Open Service Navigation Message Authentication, arXiv preprint arXiv:2501.09246, 2025.

[25] D. Q. Bui, D. T. Nguyen, and T. V. La, Research on GNSS receiver validation method based on signal simulation solution, (In Vietnamese: Nghiên cứu phương pháp kiểm định máy thu GNSS dựa trên giải pháp mô phỏng tín hiệu), Journal of Surveying and Mapping Science, no. 49, pp. 8–14, Sept. 2021. https://doi.org/10.54491/jgac.2021.49.541.

[26] NASA Earthdata, GNSS Broadcast Ephemeris Data Product. [Online] https://www.earthdata.nasa.gov/data/space-geodesy-techniques/gnss/broadcast-ephemeris-data-product.

[27] J. T. Townsend, Theoretical analysis of an alphabetic confusion matrix, Perception & Psychophysics, vol. 9, pp. 40–50, 1971.

[28] R. W. Sinnott, Virtues of the Haversine, Sky and telescope, vol. 68, no. 2, pp. 158, 1984.