

● WHITEPAPER · BLUETOOTH LE SECURITY

BLE application-layer authentication with ECDSA certificate chains on ESP32-C3

AUTHORS

Nanupatruni Mahesh Kumar

REVIEWER

Ravi Kumar Chintalapudi

VERSION

v1.1

DATE

2026-10-21

STATUS

Published

DISTRIBUTION

Public

CONTENTS

Abstract	6 Analysis and discussion
1 Introduction and objective	7 Limitations and threats to validity
2 Requirements and acceptance criteria	8 TRL assessment
3 System under test	9 Conclusions and next steps
4 Test setup and method	10 References
5 Results	Appendices

ABSTRACT

Abstract

Dharanova's hub-and-nodes sensor platform needs the hub and each sensor node to prove their identity to each other without relying on Bluetooth pairing, MAC addresses or shared secrets. This paper reports the design and lab validation of a mutual authentication handshake at the application layer, over BLE GATT, using P-256 ECDSA X.509 certificate chains issued offline and stored in each ESP32-C3 node's flash. A six-message handshake runs between a Python/BlueZ central on a Raspberry Pi Zero 2 W and two ESP32-C3-SuperMini nodes sending simulated load events. All six acceptance criteria were met in the lab, one of them pending re-verification on current firmware. Over $n = 35$ runs, scan-to-authorized latency averaged 7.394 s (p95 10.303 s). The firmware uses 54.7 % of the 1.25 MB application partition. LE Secure Connections does not work on the hub's onboard Bluetooth adapter and was removed, so the BLE link is unencrypted. A 41.36 h unattended run with two nodes delivered all 1649 events with no sequence gaps despite 471 disconnects, but that is not yet enough evidence for the $< 0.1\%$ loss target. A board swap traced one node's higher disconnect rate to the board itself. The work supports TRL 4.

KEY POINT

Key findings

- **Authentication:** every defined rejection case is refused and an authorized central completes on one connection. Criteria R1, R2 and R4 to R6 are met. R3 (time-token rejection) passed, but the clock check it relies on was made log-only on 2026-10-15 and has not been re-tested as a blocking check.
- **Latency:** 7.394 ± 0.521 s (mean, $k = 2$), p95 10.303 s, $n = 35$, measured on an earlier lab central, not on the installed Raspberry Pi hub.
- **Transport:** 0 of 1649 events lost over 41.36 h. With zero losses in 1649 events, the 95 % upper bound on the loss rate is 0.18 %, above the 0.1 % target. About 2995 loss-free events are needed to show < 0.1 %.
- **Security gap:** the BLE link carries no encryption, so payloads are readable by any BLE sniffer in range. Certificate authentication still stops an unauthorized central from operating a node.
- **Per-board instability:** after the two nodes swapped places, C5DA's disconnect rate rose 71 % while 6D1A's stayed within 3 %. The problem follows the C5DA board, not its location.
- **Readiness:** TRL 4, technology validated in the lab.

SECTION 1

Introduction and objective

Bluetooth LE link-layer pairing (LE Secure Connections) gives an encrypted link, but it does not authorize a specific identity: any host that pairs successfully is trusted. For sensor nodes deployed in commercial or safety-relevant settings, the decision to trust a hub has to be cryptographic and tied to a provisioned identity, not to a successful BLE connection.

This work tests whether mutual X.509 certificate authentication, the trust model used by TLS, can run at the application layer on a resource-constrained BLE peripheral (ESP32-C3, single-core RISC-V at 160 MHz, mbedTLS 3.x) within the limits of GATT characteristic writes and notifications. It answers four questions:

1. Can the node and the hub authenticate each other with certificate chains over one BLE connection, and does the node refuse every defined invalid case?
2. What does the handshake cost in latency, flash and non-volatile storage?
3. Once authenticated, does the BLE transport deliver every sensor event over long unattended operation with real connection drops?
4. What evidence is still missing before TRL 5 and before production?

In scope: certificate provisioning, the mutual authentication handshake, time-bounded token validation, rejection-path coverage, firmware resource use, end-to-end authentication latency, and event delivery over an unattended multi-day run with two nodes. The sensor payload is a

load value simulated in the node firmware, not a physical load cell (Table 3.1): this paper validates the authentication and transport protocol, not weight sensing.

Out of scope: more than two nodes, over-the-air certificate rotation, power profiling, secure boot, and BLE link-layer encryption. LE Secure Connections was evaluated and removed (section 6.4); it is not part of the design reported here. The BLE link is unencrypted and certificate authentication is the only trust layer.

SECTION 2

Requirements and acceptance criteria

The authentication criteria R1 to R6 were written in the first draft of this paper on 2026-10-05, after a single informal successful handshake that day and before any systematic test. The tests that check them (the Python and firmware rejection suites and the $n = 35$ latency series) all ran on 2026-10-06. The transport criteria P1 to P3 were set before the long unattended runs began.

Table 2.1: Authentication acceptance criteria

ID	Requirement	Acceptance criterion
R1	Node shall refuse events to an unauthenticated central	RESULT <code>authorized=False</code> for any connection that does not complete PROOF with a valid certificate chain
R2	No MAC address in the authorization decision	Authorization based only on certificate content; MAC not read or compared in the auth path
R3	Node shall reject an expired or replayed time token	<code>time-clock-policy</code> rejection for tokens outside a ± 30 s window or past a 120 s lifetime (<code>AUTH_TIME_MAX_LIFETIME</code> , Appendix A).
R4	Node shall reject a mismatched certificate chain	<code>certificate-rejected</code> for any chain that does not verify against the provisioned root
R5	Handshake shall complete on a single BLE connection	<code>authorized=True</code> within one <code>BleakClient</code> session, no reconnect required
R6	Firmware shall fit within 60 % of the 1.25 MB application partition	Binary size ≤ 768 KB

Table 2.2: Transport reliability criteria (Phase 5)

ID	Requirement	Acceptance criterion
P1	Sensor data is not lost in transport	Data loss $< 0.1\%$ over at least 72 h (target 168 h), measured by sequence numbers. Re-scoped on 2026-10-14 to "data can never be lost in transport"
P2	Hub recovers from every connection drop	No unrecoverable connection failure: the hub reconnects after every drop without a restart
P3	No resource leak on the hub	Hub heap stable over the run (no unbounded growth)

P1's re-scoped wording cannot be shown by any finite test. Section 6.2 proposes a statistical form of it.

SECTION 3

System under test

The system is one hub (the BLE central) and two sensor nodes (BLE peripherals). Figure 1 shows the parts and how credentials reach them; Figure 2 shows the handshake.

● SYSTEM UNDER TEST: ONE HUB, TWO NODES

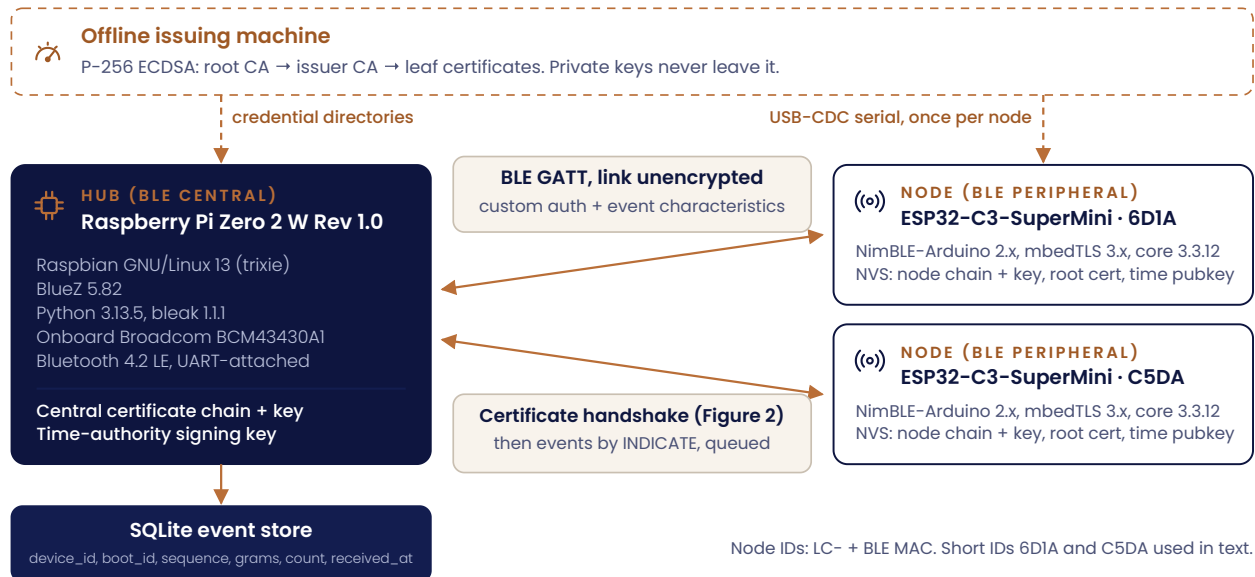


Figure 1 The hub, the two nodes and the offline credential chain.

Table 3.1: Components

Item	Detail
Node board	ESP32-C3-SuperMini, both nodes. A generic module with no printed hardware revision, so none is reported
Node IDs	6D1A and C5DA, the last four hex digits of each node's ID
Node software	NimBLE-Arduino 2.x on ESP32 Arduino core 3.3.12 (ESP-IDF 5.x, mbedTLS 3.x). Load-cell peripheral sketch
Node firmware	The 2026-10-06 build for the latency, rejection and first build-size results. The 2026-10-17 build, which sends events only once the hub has actually subscribed, for the soak run and the board swap; a later change touched only a code comment
Hub	Raspberry Pi Zero 2 W Rev 1.0, Raspbian GNU/Linux 13 (trixie), BlueZ 5.82, Python 3.13.5, bleak 1.1.1
Hub Bluetooth	Onboard Broadcom BCM43430A1, UART-attached, HCI Version: 4.2 . No USB adapter
Sensor on the node	None. <code>readSimulatedLoadGrams()</code> returns a flat 50 g and steps to <code>EVENT_GRAMS = 135 g</code> for 500 ms once every <code>EVENT_PERIOD_MS = 3 minutes</code> , crossing the <code>100 g THRESHOLD_GRAMS</code> to fire one event. A deliberate choice for exercising the authentication and transport pipeline

Certificate infrastructure. P-256 ECDSA in a three-level hierarchy: root CA, issuer CA, leaf. Each node stores its leaf and issuer certificates as one chain in NVS, its private key, the root certificate as a separate entry, and the public key of the time authority. The hub holds the same structure

for itself. Time attestations are signed with a separate P-256 key, the time-authority key, which the hub's software reads from a directory (`BLE_TIME_AUTHORITY_DIR`). All keys are generated offline and private keys never leave the issuing machine.

● APPLICATION-LAYER HANDSHAKE ON ONE BLE CONNECTION

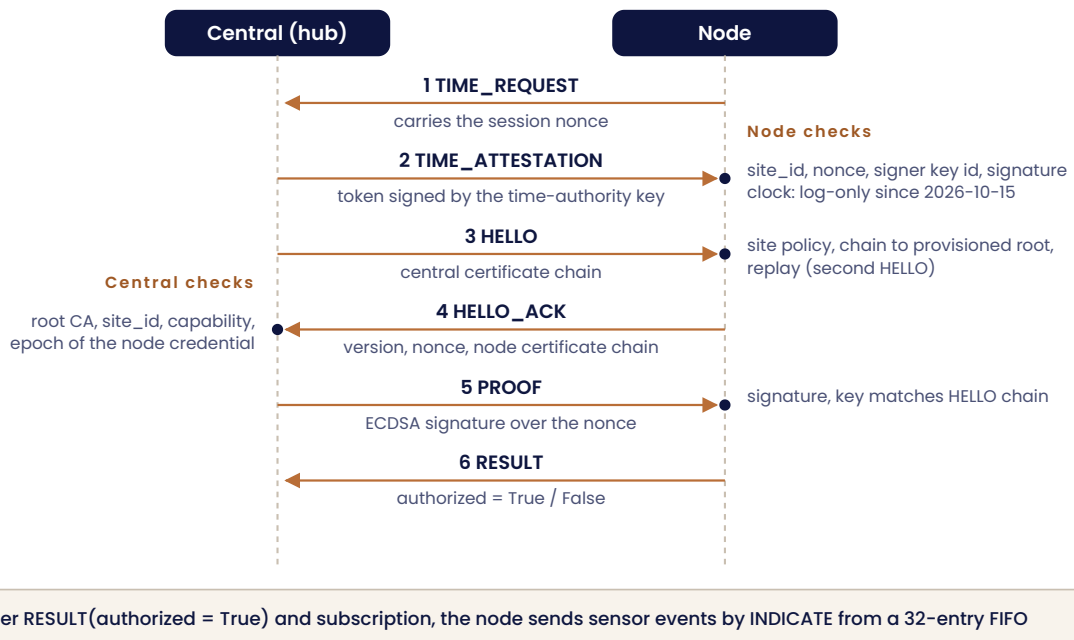


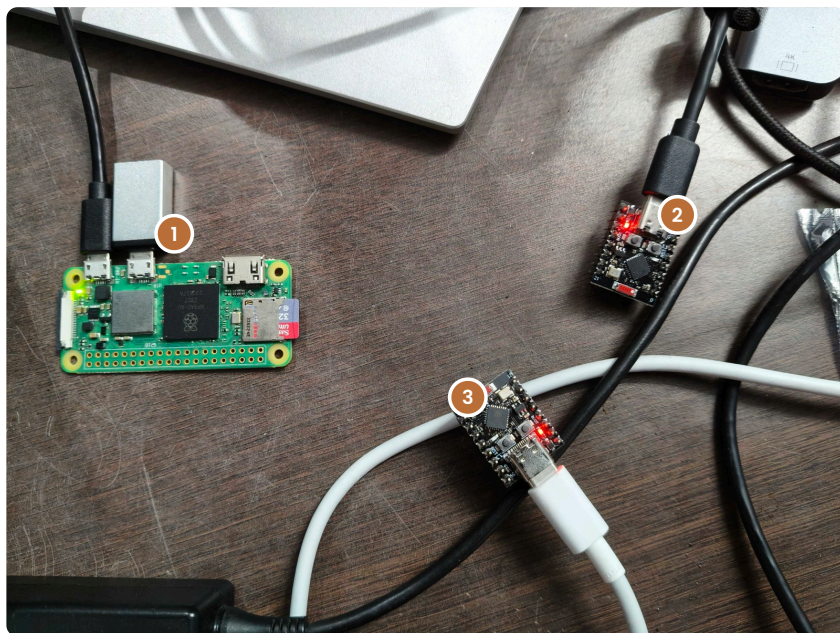
Figure 2 The six-message handshake on one connection. The checks listed beside each message are the ones the rejection tests in section 5.2 exercise. Directions and contents are taken from the hub and node source code; **HELLO_ACK** is a 1-byte version, the echoed 32-byte nonce and the node's length-prefixed certificate chain (`HELLO_ACK_HEADER = "!B32sH"`).

Provisioning. Each node is provisioned once over USB-CDC serial by a provisioning script, using a seven-step protocol: `auth-begin`, node certificate, node key, root certificate, time-key ID, time key, `auth-commit`. A `reboot` command then reloads NVS.

Event delivery. After **RESULT(authorized=True)** and subscription, the node sends its simulated load events by BLE **INDICATE**, which the hub's stack acknowledges, from a 32-entry FIFO. An event leaves the queue only once its acknowledgement arrives. Every event carries the node's `boot_id` and a per-boot `sequence` number, which is how lost events are detected. Both mechanisms are the outcome of defects found during this work (section 6.3).

Hardware as photographed. Figures 3 and 4 show the hardware, photographed in October 2026 when the hub and nodes were moved to their installed positions, the placement used for the soak run and the board swap.

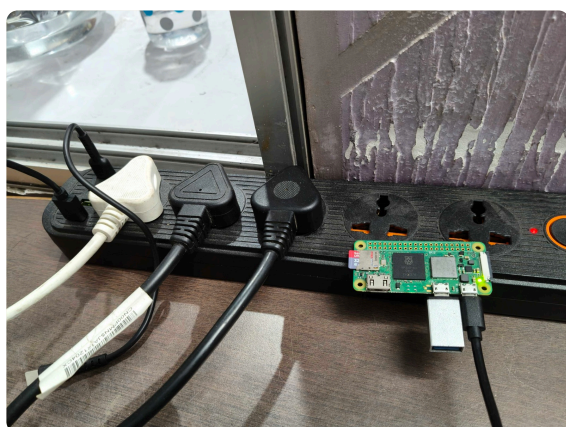
● HUB AND BOTH NODES ON THE BENCH



- 1 Hub: Raspberry Pi Zero 2 W**
microSD card fitted, powered over micro-USB
- 2 Node board**
ESP32-C3-SuperMini, powered over USB-C
- 3 Node board**
ESP32-C3-SuperMini, powered over USB-C

Figure 3 The hub and both ESP32-C3-SuperMini node boards on the bench, each powered over USB.

● HUB AND NODES AS INSTALLED

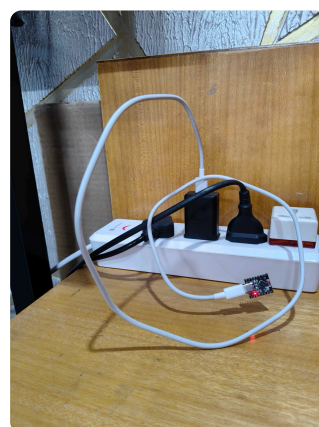


Hub

Raspberry Pi Zero 2 W on a power strip



Node 6D1A



Node C5DA

Figure 4 The hub on its mains power strip, and the two nodes as installed, each on its own USB supply: 6D1A (centre) and C5DA (right).

SECTION 4

Test setup and method

4.1 Equipment

Instrument	Model	Range / resolution used
Hub, BLE central	Raspberry Pi Zero 2 W Rev 1.0	Central application, event store, soak and swap runs
Hub Bluetooth adapter	Onboard Broadcom BCM43430A1 (UART)	Bluetooth 4.2 LE
Lab central (latency, rejection and LE SC runs, up to 2026-10-07)	Development laptop: Ubuntu 26.04.1 LTS, BlueZ 5.85, Python 3.14.4, onboard Intel AX201 Bluetooth (HCI 5.2). Software versions are as of 2026-10-19; the versions on 2026-10-06 and 10-07 were not recorded	Central application for the lab runs
Nodes	ESP32-C3-SuperMini boards 6D1A and C5DA (see Table 3.1)	Devices under test
USB cable	USB-A to USB-C	Programming and CDC serial

No calibrated instrument was used. Timing comes from host clocks and firmware log timestamps (section 5.8).

4.2 Environment

Indoor open office, unshielded. Temperature and humidity were **not instrumented**; conditions were ordinary indoor office ambient by observation, so no °C or %RH figure is reported. The office Wi-Fi is the only known concurrent 2.4 GHz source; no RF survey was done. Hub-to-node distance was **not measured**: most runs had both in the same room, roughly arm's length to a few metres apart. The one deliberate distance change (different rooms, section 6.5) was qualitative.

4.3 Procedures

The work ran as a sequence of short test phases between 2026-10-05 and 2026-10-19 (Figure 5). Defects found in one phase were fixed before the next.

● TEST CAMPAIGN, OCTOBER 2026

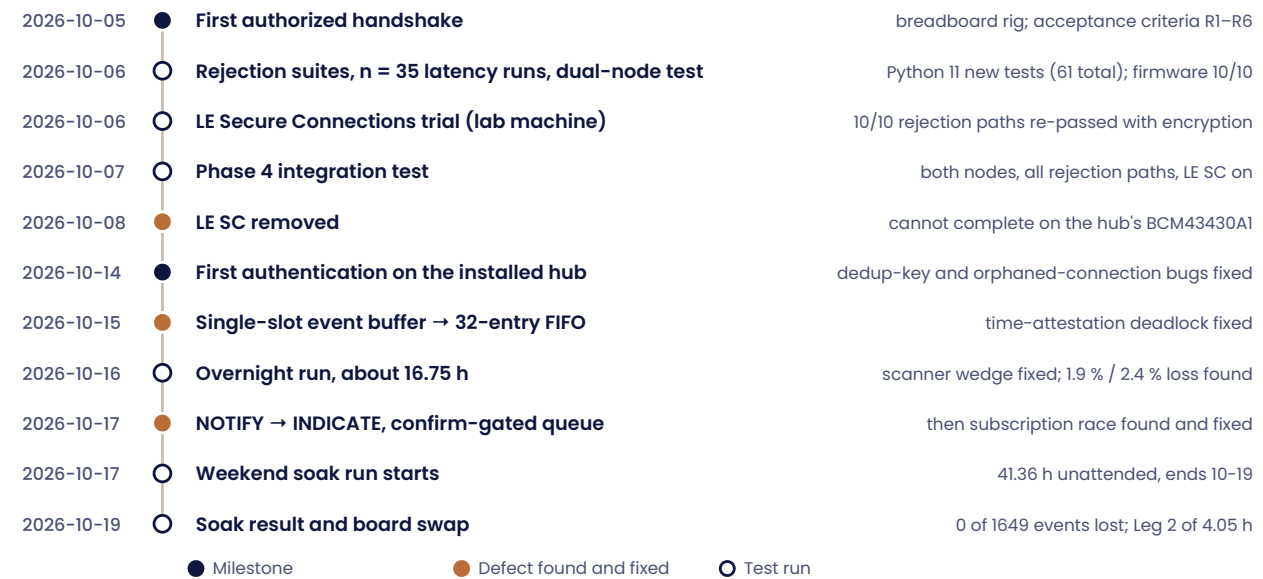


Figure 5 Test campaign, 2026-10-05 to 2026-10-19.

A. Authentication latency (2026-10-06, lab central):

- Flash the node: `arduino-cli upload -b esp32:esp32:esp32c3:CDCOnBoot=cdc -p /dev/ttyACM0`.
- Provision the node: `python3 provision_node.py /dev/ttyACM0 <node-cred-dir> <authority-dir>`, then reboot.
- Remove the stale BlueZ device cache: `bluetoothctl -- remove <MAC>`.
- Set `BLE_AUTH_ENABLED=1`, `BLE_CENTRAL_CREDENTIAL_DIR`, `BLE_TIME_AUTHORITY_DIR`, `BLE_AUTH_SITE_ID` and `BLE_TARGET_IDS`.
- Start the central (`python3 -m central`) and record the time scanning starts and the time of the certificate handshake authorized log line.
- Check `authorized=True` and `LC-<ID>` connected and subscribed in the central log.
- Repeat steps 3 to 6, 35 times.

B. Rejection paths. Python side: unit tests, including a fake node. Firmware side: a test harness sends one deliberately invalid message per scenario to node 6D1A and checks the reason the node returns. The firmware suite was re-run on the installed hub on 2026-10-14.

C. Unattended soak run. Both nodes and the hub ran without intervention from 2026-10-17T11:45:28Z to 2026-10-19T05:07:16Z. Loss was measured by a sequence-gap query on the hub's event-log table (not on the connection log). Node resets were checked by confirming that each node's `boot_id` did not change; hub restarts from the service log (`NRestarts=0`). Appendix B gives the gap query.

D. Board swap. Leg 1 is the soak run, with both boards in their usual spots in the same room. For Leg 2 the two boards were physically exchanged, leaving outlets, adapters and cables in place, and run for 4.05 h. Neither board's `boot_id` changed across the swap. Leg 2 ran from 2026-10-19T05:33:13Z to 2026-10-19T09:36:25Z. The hub's per-node counters rose from 307 to 316 for 6D1A and from 428 to 492 for C5DA over that window, which gives the Leg 2 disconnect rates in Table 5.7.

4.4 Runs and sample size

Test	n	Date	Comment
Authentication latency	35 runs	2026-10-06	Lab central, single node
Firmware rejection paths	10 scenarios, 1 pass each	2026-10-06, re-run 2026-10-14	Each scenario is a functional check, not a statistic
Python rejection paths	11 new tests, 61 total	2026-10-06	Unit tests
Dual-node concurrent authentication	1 run per configuration	2026-10-06 to 10-07	Indicative only
Unattended soak run	1 run, 1649 events	2026-10-17 to 10-19	Sample size limits the loss claim (section 6.2)
Board swap	1 swap	2026-10-19	Leg 2 lasted 4.05 h

CAUTION

Single runs

The dual-node timings, the soak run and the board swap are each a single run. They are reported as observations, and the conclusions drawn from them are limited to differences that are large compared with counting noise (section 5.8).

4.5 Standards followed

- Bluetooth Core Specification 4.2, method only: BLE transport and GATT. This is the version the hub's adapter reports. The GATT model the protocol uses is unchanged since version 4.0.
- RFC 5280, X.509 v3 certificate and CRL profile, as implemented by mbedTLS 3.x.
- ETSI EN 303 645 V3.1.3 (2024-09), provision 5.1 (no universal default passwords), method only, no deviation.
- NISTIR 8259A device cybersecurity baseline, method only, applied to the provisioning model.

No accredited or certified testing was done.

SECTION 5

Results

Values are **measured** unless marked calculated. Every calculated value, and every chart, is produced by the analysis scripts published with this paper. The soak-run, board-swap, build-size and dual-node results were recorded in the project log rather than as separate data files.

5.1 Authentication latency

Table 5.1: Scan start to `authorized`, n = 35, 2026-10-06 (calculated from the 35 recorded runs)

Statistic	Value (s)
Mean	7.394
Standard deviation	1.542
Minimum / maximum	5.772 / 11.667
p50 / p95 / p99	6.582 / 10.303 / 11.587
Standard uncertainty of the mean (Type A)	0.261
Expanded uncertainty of the mean, k = 2	0.521

● END-TO-END AUTHENTICATION LATENCY, N = 35 RUNS, 2026-10-06

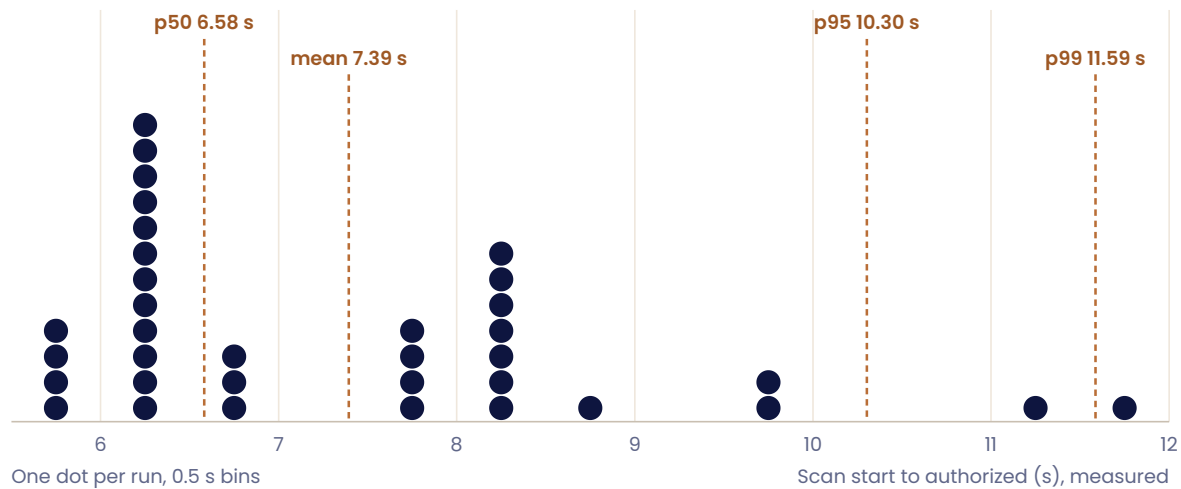


Figure 6 Distribution of the 35 latency runs. Most runs complete in 6 to 7 s, with a tail to 11.7 s.

All 35 runs succeeded and no outliers were removed. Percentiles use linear interpolation. With $n = 35$, p99 falls between the two largest values, so treat it as indicative. Each run's latency agrees with the difference of its two wall-clock timestamps to within 0.80 ms (calculated).

Latency was timed with the host's wall clock, as the CSV column names say. A wall-clock step during a run (for example an NTP correction) would distort that run's latency; none is evident in the data, but none was ruled out.

These runs used the lab central before the move to the Raspberry Pi hub. Latency on the installed hub has not been measured as a series (section 5.5 gives one reconnect).

5.2 Rejection-path coverage

Table 5.2: Firmware rejection scenarios on node 6D1A (2026-10-06; re-run 10/10 on the installed hub 2026-10-14)

Rejection reason	Trigger	Result
time-site-rejected	Time token with the wrong site_id	Pass
time-nonce-rejected	Time token with the nonce zeroed (not matching TIME_REQUEST)	Pass
time-signer-rejected	Time token with the wrong signer_key_id	Pass
time-signature-rejected	Correct signer_key_id , token signed with a different key	Pass

Rejection reason	Trigger	Result
<code>time-clock-policy</code>	Time token with <code>issued_at</code> = now + 3600 s	Pass
<code>hello-policy-rejected</code>	Valid time token, HELLO with the wrong <code>site_id</code>	Pass
<code>central-chain-verify-failed</code>	Valid time token, HELLO with a certificate from a different root CA	Pass
<code>proof-rejected</code>	Complete handshake, PROOF with the last byte of the DER signature flipped	Pass
<code>hello-replay</code>	Valid HELLO accepted, second HELLO sent before PROOF	Pass
<code>frame-header-invalid</code>	Frame with wrong magic bytes in place of TIME_ATTESTATION	Pass

Table 5.3: Python-side rejection tests (2026-10-06)

Category	Paths verified	Method
Frame violations	Unsupported protocol version, unknown message type	Unit test
Peer credential	Wrong root CA, wrong <code>site_id</code> , missing capability, stale epoch	Unit test
Handshake state machine	<code>authorized=False</code> propagation; <code>accept_result</code> in the wrong state (IDLE, after REJECTED)	Unit test
Live authenticator	Node rejection reason surfaced in <code>LiveAuthenticationError</code> ; early <code>RESULT(False)</code> before <code>HELLO_ACK</code>	Unit test with a fake node

5.3 Dual-node concurrent authentication

Both nodes authenticated at the same time over independent connections in a single `asyncio.gather` run. No cross-connection interference was seen.

Table 5.4: Dual-node runs, scan start to `authorized`, n = 1 each

Run	Link	6D1A (s)	C5DA (s)
Phase 2, 2026-10-06	Unencrypted, lab central	13.09	9.76
Phase 3, 2026-10-06	LE SC, lab central	12.46	13.07
Phase 4, 2026-10-07	LE SC	6.71	6.10

Both LE SC rows describe a configuration that was later removed (section 6.4).

5.4 Firmware and storage footprint

Table 5.5: Build output and NVS entry sizes

Item	Value	Source
Firmware binary, 2026-10-06 build	717,447 bytes	<code>arduino-cli</code> build output
Share of the 1,310,720-byte application partition	54.7 %	Calculated
Headroom to the R6 limit (768 KiB)	68,985 bytes	Calculated
Static RAM (globals), 2026-10-06 build	25,640 bytes	<code>arduino-cli</code> build output

Item	Value	Source
Firmware binary, 2026-10-17 soak-run build	718,893 bytes	<code>arduino-cli</code> , rebuilt 2026-10-19
Share of the partition, 2026-10-17 build	54.8 %	Calculated
Headroom to the R6 limit, 2026-10-17 build	67,539 bytes	Calculated
Static RAM (globals), 2026-10-17 build	26,168 bytes	<code>arduino-cli</code> , rebuilt 2026-10-19
NVS: node certificate chain	1,920 bytes	Boot log <code>nodeCert=</code> , measured
NVS: node private key	241 bytes	Boot log <code>nodeKey=</code> , measured
NVS: root certificate	546 bytes	Boot log <code>rootCert=</code> , measured
NVS: time-authority public key	178 bytes	Boot log <code>timePub=</code> , measured
NVS: all credentials	2,885 bytes	Calculated

The soak-run build, which adds the 32-entry event FIFO and the subscription-gated drain, is 1,446 bytes larger in flash and 528 bytes larger in static RAM than the 2026-10-06 build (calculated). It was rebuilt on 2026-10-19 rather than taken from the original build log, so the toolchain may differ slightly from the one that built the flashed binary.

5.5 Handshake trace from the installed system

On 2026-10-19 the hub was made to disconnect node 6D1A (`bluetoothctl disconnect`) while the node's serial output was captured (Appendix B).

● RECONNECT AND RE-AUTHENTICATION, NODE 6D1A SERIAL LOG, 2026-10-19

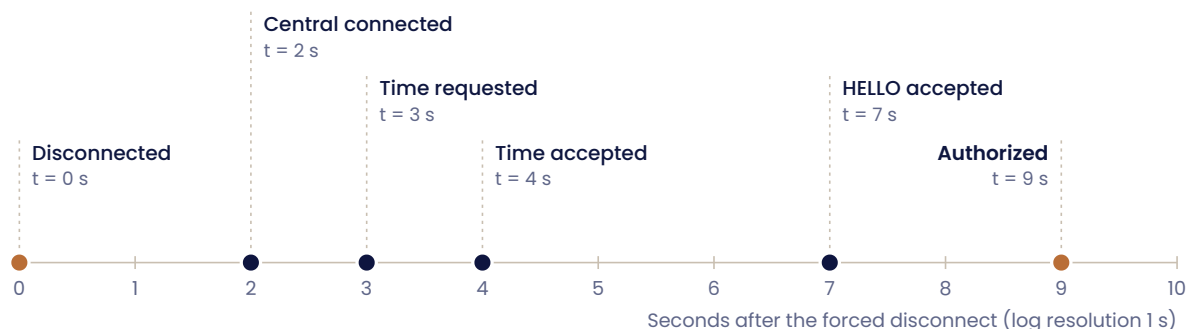


Figure 7 The node's view of one reconnect and re-authentication on current firmware, from its serial log.

Calculated from the log (1 s resolution): 9 s from the disconnect to `authorized`, of which 7 s ran from the hub reconnecting to `authorized`: 2 s to accept the time attestation, 3 s more to accept HELLO, and 2 s more to authorize. This is one reconnect, and it is timed from the disconnect rather than from scan start, so it is not directly comparable with Table 5.1.

All 425 sample lines in the capture read `50 g`, the simulated baseline (Table 3.1). No event fired during the 45 s capture.

5.6 Unattended soak run

Table 5.6: Soak run, 2026-10-17 to 2026-10-19

Node	Events received	Sequence gaps	Events lost	Disconnects	Successful reconnects
6D1A	826	0	0	89	81
C5DA	823	0	0	382	211
Total	1649	0	0	471	292

● WEEKEND SOAK RUN, 41.36 H UNATTENDED, 2026-10-17 TO 2026-10-19

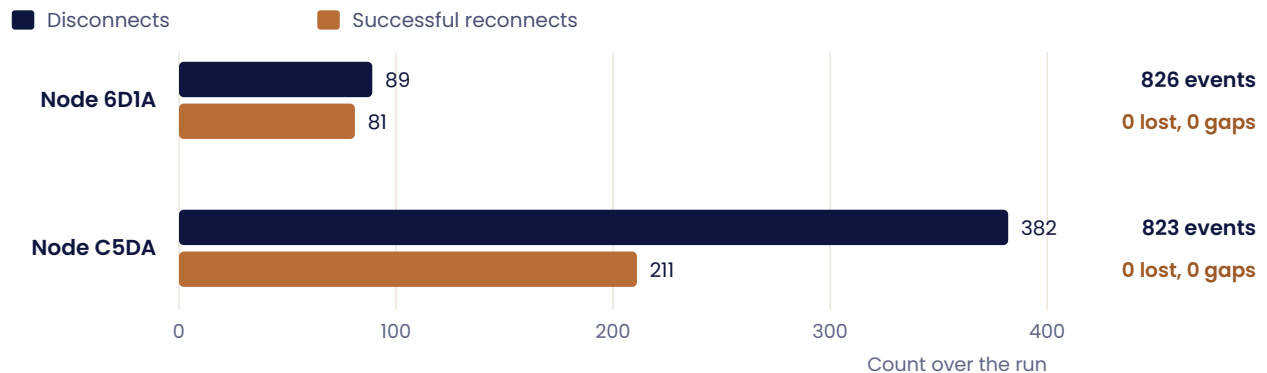


Figure 8 Disconnects and successful reconnects per node during the soak run. Every event was delivered on both nodes despite the churn.

Both nodes kept the same `boot_id` for the whole window (no resets) and the hub service logged no restarts. Calculated from these:

- The run lasted 41.363 h by its start and end timestamps, quoted as 41.36 h in this paper.
- At one event every 3 minutes, 827.3 events per node were expected. 6D1A received 1.3 fewer and C5DA 4.3 fewer. A gap query cannot see events missing at the start or end of a window. The likely explanation is that the missing events were still queued, not yet acknowledged, when the window closed: a pure window-boundary effect would be about the same for both nodes, whereas C5DA, with about 4.3 times the disconnects, has about 3.3 times the shortfall. This is inferred from the counts, not measured from the queue at that moment.
- 91 % of 6D1A's disconnects and 55 % of C5DA's were followed by a successful reconnect. As counted by the hub software, a disconnect is every exit from a node's connected block for any reason (link loss, failed authentication or any other exception, through `mark_disconnected()` in a `finally` block), and a successful reconnect is a cycle that completes both the handshake and the GATT subscription (`connected` and `subscribed`). A disconnect without a matching reconnect line does not mean lost data: a later attempt reconnects and the queue still delivers.
- The pre-fix baseline, from an overnight run of about 16.75 h on 2026-10-16/17, was 1.9 % loss on 6D1A and 2.4 % on C5DA, in 11 separate sequence gaps.

5.7 Board swap

Table 5.7: Disconnect rate before and after swapping the two boards' spots

Board	Leg 1, own spot (/h)	Leg 2, other board's spot, 4.05 h (/h)	Change (calculated)
6D1A	2.15 ± 0.23	2.22 ± 0.74	+3 %
C5DA	9.24 ± 0.47	15.79 ± 1.97	+71 %

● **BOARD SWAP: DISCONNECT RATE FOLLOWS THE BOARD, NOT THE SPOT**

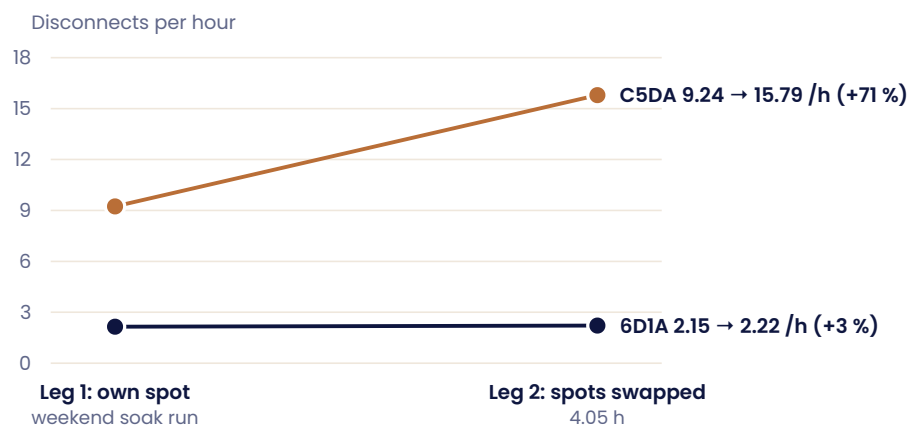


Figure 9 Disconnect rate per board before and after the swap, calculated from the soak-run counts (Leg 1) and the hub's counter deltas (Leg 2).

The $\pm$ values are one standard uncertainty from counting statistics ($\sqrt{\text{count}} / \text{duration}$, calculated). Leg 2 lasted 4.053 h by its timestamps, with counter deltas of 9 (6D1A) and 64 (C5DA), which reproduce the reported rates. If C5DA's Leg 1 rate had held, about 37.4 disconnects would have been expected in Leg 2; the chance of 64 or more under that rate is 0.000049 (one-sided Poisson, calculated). 6D1A's change is well within counting noise ($p = 0.51$).

Events during Leg 2. Both nodes delivered their events with no sequence gaps, including C5DA at its higher disconnect rate.

5.8 Uncertainty

- **Latency, Type A:** standard uncertainty of the mean 0.261 s; expanded 0.521 s at $k = 2$ (about 95 %). Run-to-run spread ($s = 1.542$ s) is far larger than any timing-resolution effect.
- **Latency, Type B:** host wall-clock resolution 1 ms in the CSV, negligible, assuming no clock step during a run (section 5.1).
- **Handshake trace:** log timestamps have 1 s resolution, so each interval in section 5.5 is uncertain by up to ± 1 s.
- **Soak counts:** exact counts from the event log; the uncertainty is in what they show about the loss rate (section 6.2), not in the counts.
- **Disconnect rates:** Poisson counting uncertainty as in Table 5.7. Leg 2 counts are counter deltas.
- **Build and NVS sizes:** exact.

SECTION 6

Analysis and discussion

6.1 Against the authentication criteria

ID	Criterion	Result	Met?
R1	Reject an unauthenticated central	An unprovisioned central receives <code>certificate-rejected</code> ; all 10 firmware scenarios refused	Met
R2	No MAC in the auth decision	Auth path uses certificate content only; MAC never read in that path (code inspection)	Met
R3	Reject expired or replayed tokens	<code>time-clock-policy</code> fired as expected on 2026-10-06 and 2026-10-14. On 2026-10-15 the clock check was moved after signature verification and made log-only (section 6.3), and the scenario has not been re-run as a blocking check since	Partly met (re-test on current firmware)
R4	Reject a mismatched chain	<code>central-chain-verify-failed</code> for a certificate from another root CA	Met
R5	One-connection handshake	<code>authorized=True</code> without reconnect, first on 2026-10-05 and in all 35 latency runs	Met
R6	Binary ≤ 768 KB	717,447 bytes (2026-10-06 build) and 718,893 bytes (2026-10-17 soak-run build), 54.7 % and 54.8 % of the partition	Met

R4's criterion names the reason `certificate-rejected`, but the firmware reports `central-chain-verify-failed` for this case. The behaviour meets the intent; the criterion text should be aligned with the firmware's reason codes.

6.2 Against the transport criteria

ID	Criterion	Result	Met?
P1	Loss < 0.1 % over ≥ 72 h	0 of 1649 events lost over 41.36 h. Duration short of 72 h, and the sample cannot show < 0.1 % (below)	Partly met (zero loss seen, target not demonstrated)
P2	Recover from every drop	No hub restarts, no node resets, events delivered through 471 disconnects	Met
P3	Hub heap stable	Not recorded	Not met

● WHAT ZERO LOSS EVENTS CAN SHOW: 95 % UPPER BOUND ON THE LOSS RATE

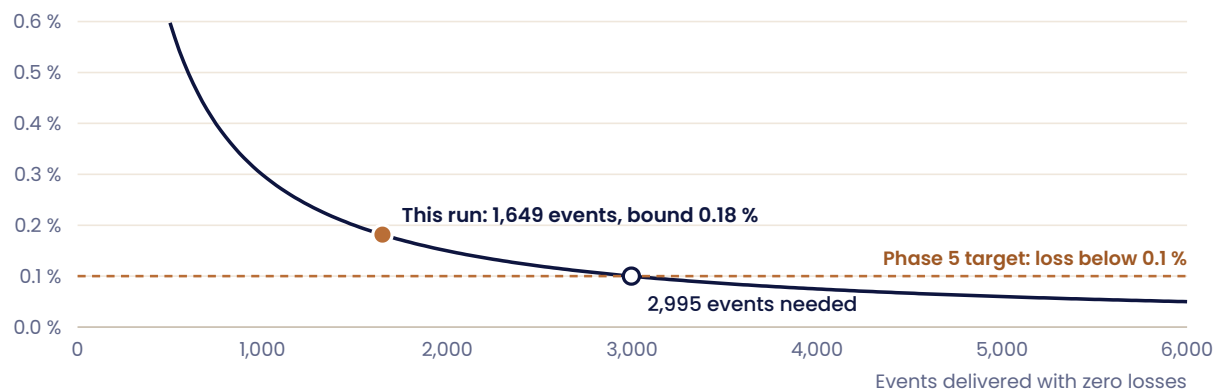


Figure 10 The 95 % upper bound on the true loss rate after observing no losses, as a function of the number of events delivered (exact binomial bound, $1 - 0.05^{1/n}$).

Zero observed losses do not mean a zero loss rate. With 0 losses in 1649 events, the loss rate can only be said to be below 0.18 % at 95 % confidence (calculated), which is above the 0.1 % target. Showing < 0.1 % needs at least 2995 loss-free events, about 74.9 h with two nodes at one event every 3 minutes. Even a complete 72 h run with two nodes (2880 events) would give a bound of 0.104 %, just short. **PI should be restated** as a statistical criterion, for example "no losses in at least 3000 consecutive events across at least 72 h", and run with more nodes or a faster event rate to reach it sooner. The re-scoped "never lost in transport" goal is a design property (every event is queued until acknowledged), best argued from the design and tested by fault injection, as in sections 6.3 and 6.5, alongside the statistical run.

The improvement over the pre-fix baseline is clear: at 1.9 % to 2.4 %, about 31 to 40 losses would have been expected in 1649 events, and none occurred.

6.3 Defects found by running on the real hardware

Most of the defects below appeared only when the system ran unattended for hours on the installed hub. None was caught by the rejection suites or by short bench runs.

Date	Defect	Effect	Root cause	Fix
10-06	Node key silently rejected	Persistent node-key-mismatch despite correct NVS contents	mbedtls 3.x needs a non-null <code>f_rng</code> for <code>mbedtls_pk_parse_key</code> and <code>mbedtls_pk_check_pair</code> ; <code>nullptr</code> fails without a compile-time warning	Pass an <code>esp_fill_random</code> (hardware TRNG) wrapper
10-06	Board C5DA not authenticating	<code>authProvisioned</code> stayed false although credentials were stored	Older firmware binary predating the <code>f_rng</code> fix	Reflash; stored credentials then loaded
10-14	Event dedup key	Database stuck on one row for two weeks	Key <code>(device_id, sequence)</code> ; <code>sequence</code> resets on reboot	Key <code>(device_id, boot_id, sequence)</code>

Date	Defect	Effect	Root cause	Fix
10-14	Orphaned BLE connection	After <code>systemctl restart</code> , link left connected with no owner	No SIGTERM/SIGINT handler	Signal handler; also fixed <code>CentralApp.stop()</code>
10-15	Single-slot event buffer	Events lost across a disconnect longer than one 3-minute cycle	Pending-event buffer held one event	32-entry FIFO; each event keeps its sequence number
10-15	Time-attestation deadlock	After repeated warm resets, node rejected every attestation until power-cycled (27 min reproduced)	Clock checked against the node's stale RTC before signature verification	Verify signature first; clock check log-only, a valid attestation corrects the clock
10-16	Hub scanner wedge	Both nodes dark for over 2 h; 1,120 consecutive identical failures	Scanner restarted BlueZ discovery about every 6 s; one failed stop cascaded	Persistent scanner, liveness watchdog, unprivileged adapter power-toggle, then process exit under <code>systemd</code> <code>Restart=always</code>
10-17	Unconfirmed NOTIFY fast path	1.9 % / 2.4 % loss overnight	Connected events sent by NOTIFY, bypassing the queue, and lost when the link dropped	Always queue; send by INDICATE; dequeue only on acknowledgement
10-17	Inverted status check in that fix	6D1A stuck retrying event 0	NimBLE returns 0 for "sent" and BLE_HS_EDONE (14) for "acknowledged"; the two were swapped	Corrected; found from live serial output
10-17	Subscription race	Delivery stalled 81+ min on 6D1A and 5+ h on C5DA while connections still succeeded	Queue drained when <code>centralConnected</code> was set, before the hub had subscribed; the stack gives no completion callback in that case	Gate the drain on the characteristic's real subscription state

The last three rows show why a passing stress test and a permanent fix are different claims. The NOTIFY fix passed a dense forced-disconnect campaign with zero gaps, and hours later a different timing stalled delivery on both nodes. The stall was diagnosed the same day from live serial output, fixed, and re-verified on both boards before the soak run started.

6.4 Link-layer encryption: evaluated and removed

LE Secure Connections (Just Works, NoInputNoOutput) was enabled on both nodes and passed on the lab central on 2026-10-06: all 10 rejection scenarios still passed, and the application-layer logic was unaffected by the encrypted link. On 2026-10-08, on the Raspberry Pi hub, the onboard BCM43430A1 could not complete LE SC at all: with `setSecurityAuth(BLE_SM_PAIR_AUTHREQ_SC)` NimBLE rejected the CCCD write with "Insufficient Authentication", giving a 2 s pairing timeout before every disconnect.

LE SC was removed rather than worked around. Certificate authentication is now the only trust layer and the link is unencrypted. This keeps unauthorized centrals from operating a node, but provides **no payload confidentiality**: any BLE sniffer in range can read sensor events. The LE SC results in Table 5.4 do not apply to the installed system. Either a link-layer scheme that works with this adapter or application-layer payload encryption is needed before production (section 7.2).

6.5 Placement against per-board instability

C5DA disconnected about 4.3 times as often as 6D1A during the soak run (calculated). Earlier tests left the cause open. With the nodes in different rooms, C5DA logged 16 disconnects in 12 minutes with RSSI down to -97 dBm, which suggested distance. With the nodes co-located, C5DA's RSSI still swung between -99 and -79 dBm within a minute, next to 6D1A at a steady -75 to -90 dBm.

The swap separates the board from the spot. If position caused the drops, 6D1A should have worsened in C5DA's old spot and C5DA should have improved in 6D1A's. Instead, 6D1A held within counting noise and C5DA got worse. The instability follows the C5DA board (its radio, antenna or module). Three cautions apply:

- Leg 1 (weekend) and Leg 2 (Monday, 4.05 h) ran at different times, so office occupancy and Wi-Fi traffic differed. This may explain why C5DA got *worse* rather than staying level, but not why 6D1A was unaffected.
- It is one swap. Swapping back (Leg 3), or adding a third board, would rule out a time effect.
- C5DA is the board that was reflashed after an earlier physical mix-up. Both boards are the same model (ESP32-C3-SuperMini); with no printed revision, a difference in production batch cannot be ruled out.

Neither node showed a sequence gap during Leg 2, so the confirm-gated queue held under C5DA's higher disconnect rate.

6.6 Notes for repeating this work

- `ESP.restart()` keeps the ESP32-C3 RTC; only a power-on reset clears it. A test that sets the clock from a token affects every later run until the board is power-cycled.
- USB-CDC serial output on the ESP32-C3 is unreliable during BLE activity. Messages printed from BLE callbacks may never reach the host; boot-time logging before the BLE stack starts is reliable. Opening the port must leave DTR and RTS untouched, or the chip resets.
- BlueZ caches device state across sessions. `BleakScanner.discover()` with a UUID filter returns nothing when a device is cached as connected; run `bluetoothctl -- remove <MAC>` before each test.
- The nodes advertise with an empty name. Find them by MAC (chip MAC + 2) or by service UUID.

SECTION 7

Limitations and threats to validity

7.1 Limits of the evidence

- **Latency measured on a different central.** The $n = 35$ series used the lab central (a laptop with an Intel AX201, HCI 5.2) before the move to the Raspberry Pi hub (BCM43430A1, HCI 4.2); on the installed hub only a single reconnect was timed. The lab central's software versions at test time were not recorded. The dual-node timings are $n = 1$.
- **Firmware changed after the criteria were verified.** R1 to R5 were verified on firmware up to 2026-10-14. The event path (FIFO, INDICATE, subscription gate) and the time-attestation clock check changed afterwards. The rejection suite has not been re-run on the soak-run firmware.
- **Transport evidence is one run.** One 41.36 h run, short of the 72 h target and too small to show $< 0.1\%$ loss (section 6.2). Hub heap, node heap, reconnect latency and RTC drift were not recorded.
- **Raw output not saved.** The soak, swap, build and dual-node numbers were recorded in the project log; the query output and build logs behind them were not saved, and the gap query in Appendix B was written out afterwards from the table schema.
- **Lab environment only.** One indoor office, no RF characterisation, temperature and humidity not recorded, distances not measured.
- **Two boards, one of which misbehaves.** Both are generic ESP32-C3-SuperMini modules with no revision marking. Unit-to-unit variation is unknown.
- **Simulated sensor.** Events come from a firmware-generated load value. Real sensor timing, noise and power draw were not exercised.
- **One hub.** One Raspberry Pi with one BlueZ version and its onboard adapter.

7.2 Production-hardening gaps

These do not affect the TRL 4 claim. They are listed in order of security impact.

Gap	Impact	Notes
No secure boot	High: physical access bypasses the auth stack	Any firmware can be flashed over USB. Needs ESP32-C3 eFuse configuration and a signed binary chain
No certificate rotation	High: certificates expire 2027-10-05	No OTA path to renew node credentials. An expired node certificate means the node cannot authenticate
Unauthenticated USB provisioning	High: physical access allows re-provisioning	An attacker with USB access can install their own credentials. Lock provisioning behind secure boot
No BLE link encryption	High: payloads readable in range	See section 6.4
Time-authority key on a development machine	Medium	Needs restricted permissions and audit, or an HSM

Gap	Impact	Notes
Lab environment only	Medium	Temperature, humidity and RF not characterised

SECTION 8

TRL assessment

TECHNOLOGY READINESS: WHERE THIS WORK SITS

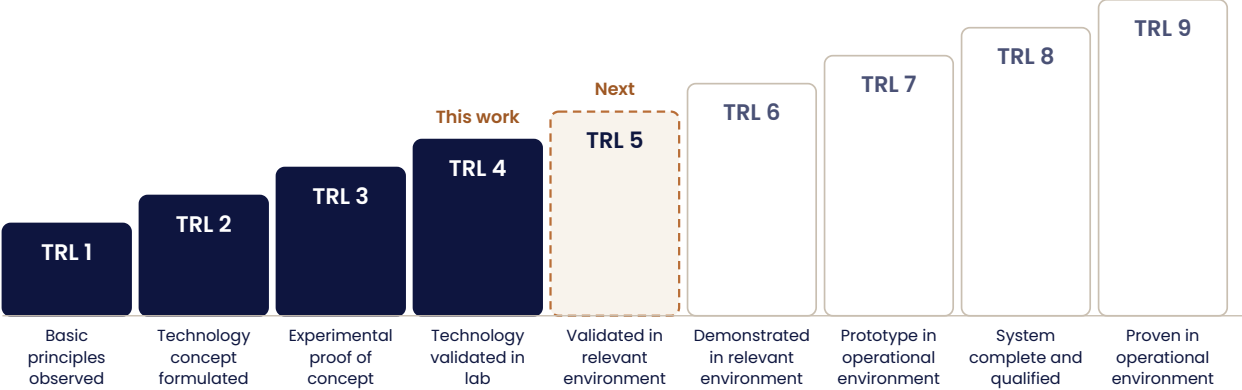


Figure 11 Where this work sits on the EU Horizon TRL scale. Definitions from the Dharanova publishing guidelines (EU Horizon TRL scale, ISO 16290 criteria).

TRL demonstrated	TRL 4, technology validated in lab, for the authentication function
Integration level	Subsystem: one hub and two nodes with the event store, authenticating and delivering events end to end
Fidelity	Breadboard: off-the-shelf ESP32-C3-SuperMini modules with a simulated sensor, USB power, no enclosure or production PCB
Environment	Lab: indoor office, uncontrolled RF, ambient temperature not recorded
Evidence	Section 5 (latency, rejection paths, footprint, soak run); section 6.1 (R1 to R6)
Caveats on the claim	R3 to be re-tested on current firmware; latency to be re-measured on the installed hub
Gaps to TRL 5	(1) Test in the relevant environment defined below, across separate rooms with representative distances and walls. (2) Re-run the rejection suite and $n \geq 30$ latency runs on the installed hub with current firmware. (3) Restate P1 statistically and meet it (at least 2995 loss-free events), and record hub heap. (4) Specify the temperature range and enclosure, which the definition below leaves open. (5) More than two node boards, to separate board defects from design issues
Gaps beyond	TRL 6 needs a representative integrated prototype (production-like board, enclosure, power, real sensor) demonstrated end to end in the relevant environment, with the section 7.2 security gaps closed or accepted

Relevant environment (for the TRL 5 claim): an indoor residential or office building, with one hub serving several sensor nodes (initially two) placed in separate rooms, so with representative multi-room distances and walls rather than one shared room as tested here. The trust

requirement is unchanged: only a central with a valid, certificate-authorized identity may connect to and exchange data with any node, in any room. Temperature range and enclosure are not yet specified.

SECTION 9

Conclusions and next steps

Mutual certificate-based authentication runs end to end on an ESP32-C3 within the BLE GATT model, on one connection, in about 7.4 s on average, with every defined invalid case refused and the firmware using just over half of its partition. The removal of LE Secure Connections leaves the link unencrypted, which is now the largest open security gap. Long unattended running found ten defects that short tests had missed; after the fixes, a 41.36 h run lost no events through 471 disconnects. That result is strong evidence of improvement but not yet proof of the < 0.1 % target. One node's instability is traced to the board. The evidence supports TRL 4.

Next steps, in priority order:

1. **Re-verify on current firmware.** Re-run the 10 firmware rejection scenarios on the 2026-10-17 build, R3 in particular.
2. **Re-measure latency on the installed hub**, $n \geq 30$, using a monotonic clock.
3. **Restate PI statistically**, then run a soak of at least 2995 events (about 75 h with two nodes), saving the gap-query output and recording hub heap, node heap, reconnect latency and RTC drift.
4. **Finish the board investigation:** swap back (Leg 3) or add a third board.
5. **Close the security gaps** in section 7.2: payload or link encryption, secure boot, locked provisioning, certificate rotation, and time-authority key management.
6. **Specify temperature range and enclosure** to complete the relevant-environment definition (section 8).
7. **Save raw evidence with each run:** build logs, query outputs and serial captures, alongside the publication data.

SECTION 10

References

1. Bluetooth SIG, *Bluetooth Core Specification*, version 4.2, 2014.
2. D. Cooper et al., *RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, IETF, 2008.
3. ETSI EN 303 645 V3.1.3 (2024-09), *Cyber Security for Consumer Internet of Things: Baseline Requirements*.
4. NIST, *NISTIR 8259A: IoT Device Cybersecurity Capability Core Baseline*, 2020.
5. Arm / Mbed TLS, *Mbed TLS 3.x API documentation*: `mbdtdls_pk_parse_key` , `mbdtdls_pk_check_pair` .

6. Espressif Systems, *ESP32-C3 Technical Reference Manual*, version 1.4 (no date printed on the cover).
7. JCGM 100:2008, *Evaluation of measurement data: Guide to the expression of uncertainty in measurement* (GUM).
8. ISO 16290:2013, *Space systems: Definition of the Technology Readiness Levels (TRLs) and their criteria of assessment*.

● APPENDICES

Appendices

A. Full configuration

Firmware build:

- FQBN `esp32:esp32:esp32c3:CDC0nBoot=cdc`, Arduino core 3.3.12, NimBLE-Arduino 2.x, mbedTLS 3.x (bundled with ESP-IDF 5.x), optimisation default (`-Os`).

Key firmware constants:

Constant	Value
<code>AUTH_TIME_CLOCK_TOLERANCE</code>	30 s
<code>AUTH_TIME_MAX_LIFETIME</code>	120 s
<code>AUTH_MAX_CHUNKS</code>	64
<code>AUTH_FRAME_HEADER_BYTES</code>	26

Certificates: ECDSA P-256 (secp256r1), X.509 v3 in PEM, chain depth 3 (root, issuer, leaf). The node stores leaf and issuer as one chain and the root separately.

B. Data and test logs

The data published with this paper:

- **Latency runs:** start and authorized timestamps and the latency of each of the 35 runs on 2026-10-06 (section 5.1).
- **Reconnect capture:** the node's serial log from the forced reconnect on 2026-10-19 (section 5.5, excerpt below).
- **Project log extracts:** the soak-run, board-swap, build-size and dual-node results.
- **Processed tables:** the latency statistics, the handshake trace with timings, and every calculated value with its formula.

The serial capture is from node 6D1A on 2026-10-19 (local time), taken with `pyserial` leaving DTR and RTS untouched. The original 2026-10-05 first handshake was observed live but never saved. The authentication lines from the capture, without the 10 Hz sample lines:

Time	Node log line
15:49:31	Central disconnected, reason=531; advertising again
15:49:33	adv-watchdog: restarting advertising
15:49:33	Auth status: provisioned;time-pending
15:49:33	Central connected; awaiting trusted time
15:49:34	Auth status: provisioned;time-requested
15:49:35	Auth status: provisioned;time-accepted
15:49:35	Trusted time attestation accepted
15:49:38	Auth status: hello-accepted;proof-pending
15:49:40	Auth status: authorized
15:49:40	Authorized central certificate handshake completed

Soak-run gap query. Written against the `loadcell_events` schema (`device_id`, `boot_id`, `sequence`, `UNIQUE(device_id, boot_id, sequence)`). The result recorded in the project log on 2026-10-19 is 826 and 823 events with no gaps (Table 5.6); the query output itself was not saved.

```
sql SELECT device_id, boot_id, COUNT(*) AS events_received, SUM(gap_size) AS events_lost FROM (
SELECT device_id, boot_id, sequence, sequence - LAG(sequence) OVER ( PARTITION BY device_id, boot_id
ORDER BY sequence) - 1 AS gap_size FROM loadcell_events WHERE received_at BETWEEN '2026-10-
17T11:45:28Z' AND '2026-10-19T05:07:16Z' ) GROUP BY device_id, boot_id;
```

As written, this query counts gaps only between received events in the window. It cannot detect events missing before the first or after the last one, which is the case discussed in section 5.6.

C. Revision history

All versions by Nanupatruni Mahesh Kumar.

Version	Date	Changes
v1.1	2026-10-21	Wording and reference corrections; results unchanged
v1.0	2026-10-20	First published version, reviewed by Ravi Kumar Chintalapudi. Released as Public
v0.19	2026-10-20	Board-swap rates and significance test computed from the hub's counter values; soak-run firmware footprint added to Table 5.5 and R6 assessed for both builds
v0.18	2026-10-19	Node photos mapped to node IDs; board-swap Leg 2 timings added; soak-run gap query added to Appendix B
v0.17	2026-10-19	Node board corrected to ESP32-C3-SuperMini (was DevKitM-1); simulated sensor payload documented; R3 token lifetime corrected to 120 s to match firmware; latency clock corrected to wall clock; handshake message contents and lab central documented; disconnect and reconnect definitions added; relevant environment for TRL 5 defined; standards references updated
v0.16	2026-10-19	Results sections, defect table and test-campaign timeline added; latency statistics, diagrams, charts, hardware photos and the loss-rate upper bound added
v0.15	2026-10-19	Board swap closes the placement question

Version	Date	Changes
v0.14	2026-10-19	Weekend soak run: 0 of 1649 events lost over about 41.4 h; Phase 5 closed short of 72 h
v0.13	2026-10-17	Subscription race root-caused and fixed
v0.12	2026-10-17	Delivery stall reported, left open
v0.11	2026-10-17	Overnight run: scanner wedge fixed; transport loss found, fixed with INDICATE
v0.10	2026-10-15	FIFO queue and time-attestation fixes; printf-blocking hypothesis refuted
v0.9	2026-10-14	First authentication on the installed hub; dedup and orphaned-connection fixes
v0.8	2026-10-08	LE Secure Connections removed
v0.7	2026-10-07	Phase 4 integration test
v0.6	2026-10-06	LE SC enabled; rejection suite re-passed with encryption
v0.5	2026-10-06	Dual-node concurrent authentication
v0.4	2026-10-06	n = 35 latency runs
v0.3	2026-10-06	Firmware rejection scenarios 10/10
v0.2	2026-10-06	Python rejection-path tests
v0.1	2026-10-05	First draft after the first handshake



We build intelligence into everyday objects.

● HOW TO CITE

Nanupatruni Mahesh Kumar. *BLE application-layer authentication with ECDSA certificate chains on ESP32-C3*. Dharanova whitepaper, v1.1, 2026.

dharanova.com · info@dharanova.com