PahiLabs

MAIA-IOT benchmark white paper

Why clients can trust MAIA-IOT device authentication.

A public-safe benchmark interpretation for buyers evaluating device trust, secure ingest, repeated operation, and rejection behavior.

Download PDF

Back to resources

Benchmark evidence snapshot

What buyers should notice first.

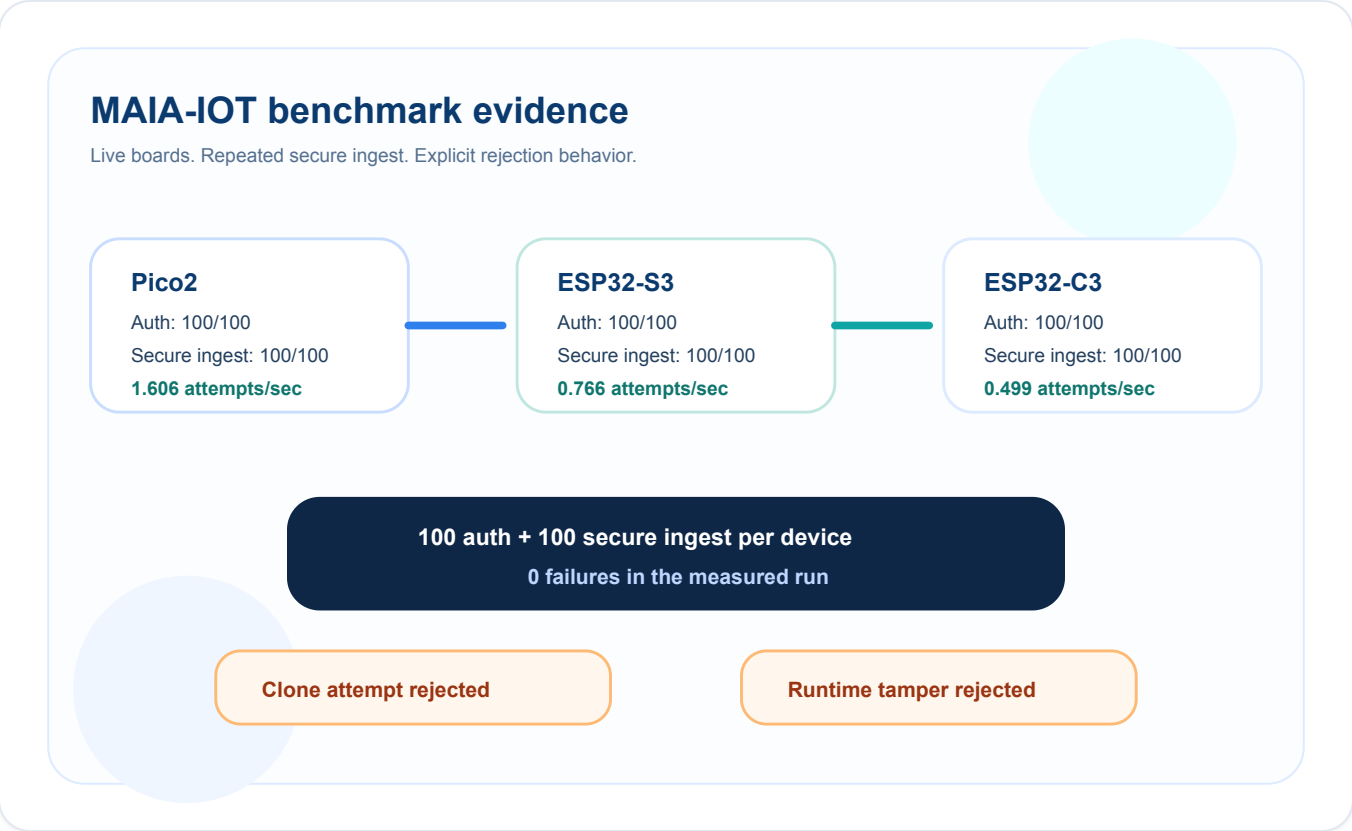
Validated on Pico2, ESP32-S3, and ESP32-C3 device classes using live device-to-server operation.

Each device completed 100/100 authentication operations and 100/100 secure ingest operations in the measured run.

Each device completed one rehandshake during the run, with zero benchmark failures reported.

Clone and runtime-tamper scenarios were rejected in the measured validation package.

The benchmark should be read as repeatability and trust-behavior evidence, not as a generic speed contest.



Executive summary

MAIA-IOT is positioned as a device trust layer for IoT, gateway, customer-premises equipment, and edge deployments where static device secrets are no longer enough. This white paper explains why a potential client should trust the technology at the evaluation stage. It does not disclose proprietary protocol details, internal implementation paths, secret material, or low-level design mechanics. Instead, it focuses on what a buyer should be able to verify: the system was exercised on real device classes, the secure workflow repeated successfully, the server made clear acceptance and rejection decisions, and the benchmark results can be interpreted in a practical deployment context.

The benchmark package measured three device classes: Pico2, ESP32-S3, and ESP32-C3. In the reported validation run, each device completed 100 authentication operations and 100 secure ingest operations. Each device also performed one rehandshake during the run. The validation summary reports zero failures for these measured paths. Throughput differed by board, which is expected because the tested hardware classes do not have identical runtime characteristics. The important trust signal is not that every board had the same speed; it is that each board completed the same security workflow and preserved the same acceptance behavior under repeated use.

For clients, the most important result is explicit rejection behavior. The validation package reports rejection of a cloned ESP32-C3 image due to hardware identity mismatch and rejection of a tampered ESP32-C3 runtime due to attestation mismatch. Those outcomes are central to the trust story. Many IoT security claims stop at encryption or connectivity. MAIA-IOT is stronger when it can show that the server accepts the right device state and rejects the wrong one. That is the difference between a communication feature and a trust product.

Why IoT buyers need benchmark evidence

IoT buyers face a practical problem: connected devices often live outside the clean boundaries of enterprise IT. They run in field locations, customer premises, industrial environments, labs, vehicles, buildings, and gateway networks. They may be updated irregularly, touched by third parties, or cloned during development and maintenance. A security control that only works on a slide is not useful in that environment. Buyers need evidence that a trust workflow can repeat, survive state changes, and distinguish a valid device from a copied or modified one.

Traditional security language often emphasizes encryption, but encryption by itself does not answer the operational trust question. A buyer still needs to know which device sent the data, whether the device is in an expected state, whether stale traffic was accepted, whether the server advanced trust state correctly, and whether a copied board image can impersonate another device. A benchmark should therefore measure more than speed. It should show repeatability, state behavior, rejection behavior, and a path toward deployment.

The MAIA-IOT validation package is useful because it turns the trust story into visible evidence. The benchmark does not ask the reader to accept a broad claim that the technology is secure. It shows repeated live operations on named device classes and separates the interpretation of throughput from the interpretation of correctness. That separation is important. A client can evaluate whether the performance profile fits their use case while also seeing that the acceptance and rejection model is deliberate.

Benchmark environment and tested devices

The measured run used live boards against a Raspberry Pi server path. The device classes listed in the validation package are Pico2, ESP32-S3, and ESP32-C3. These are practical targets for evaluation because they represent small-board and MicroPython-capable environments that buyers can understand. The

benchmark therefore avoids a common credibility problem: a system that looks strong only when tested on a powerful desktop or a synthetic endpoint. MAIA-IOT is presented as an IoT trust layer, so the evidence needs to include device classes that resemble the kinds of boards a client may actually use in pilots.

The benchmark measured authentication and secure ingest together. That matters because real device trust is not only a login event. A device usually needs to send telemetry, operational status, event data, sensor data, or gateway messages after it has been recognized. Measuring secure ingest alongside authentication gives the reader a clearer picture of the workflow. It shows that the system is not limited to a one-time handshake or enrollment demo. It can keep operating across repeated accepted messages.

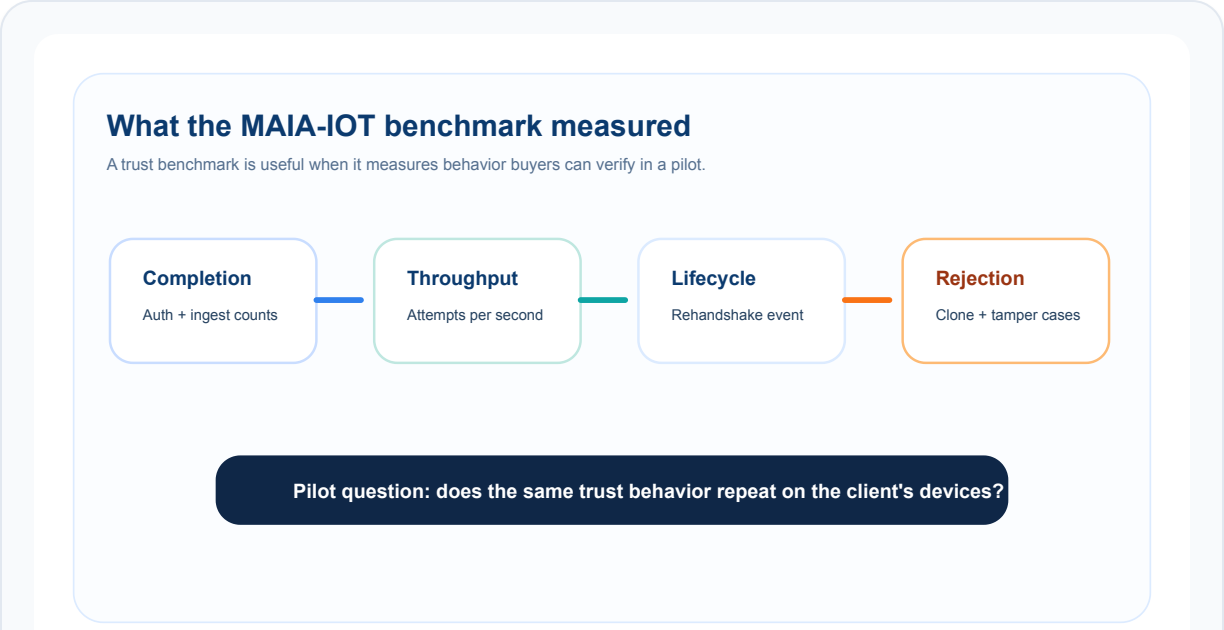
The validation package also includes one rehandshake during the run. Rehandshake behavior matters because any live system needs lifecycle events. Devices may reconnect, refresh state, recover from network changes, or continue operating across a long session. A benchmark that includes a lifecycle event is more useful than a benchmark that only repeats identical calls. It gives the buyer a better basis for asking how the system will behave during a pilot.

Metrics reported in the validation package

The public-safe metrics that should be shared on the website are straightforward. Pico2 completed 100 authentication operations, 100 secure ingest operations, one rehandshake, zero failures, and approximately 1.606 attempts per second in the measured run. ESP32-S3 completed 100 authentication operations, 100 secure ingest operations, one rehandshake, zero failures, and approximately 0.766 attempts per second. ESP32-C3 completed 100 authentication operations, 100 secure ingest operations, one rehandshake, zero failures, and approximately 0.499 attempts per second.

These numbers should be presented with care. They are not a universal performance guarantee. They are evidence from a measured validation run. Hardware class, firmware context, transport conditions, server configuration, message size, and deployment architecture can influence runtime behavior. The correct marketing message is not that every deployment will run at the exact same rate. The correct message is that three different device classes completed the same trust workflow repeatedly and that the results were summarized in a way buyers can review.

Attempts per second is useful because it gives a buyer a sense of practical runtime cost. However, the main trust metrics are completion, repeatability, rehandshake continuity, and rejection behavior. In a device trust product, correctness is more important than benchmark bragging. A faster system that accepts cloned or stale traffic is not a stronger trust layer. The validation package is valuable because it presents speed in the context of security outcomes.



Benchmark metrics used for buyer evaluation

This visual summarizes the public-safe metrics that matter: repeated completion, throughput, lifecycle behavior, and rejection outcomes.

How to interpret zero failures

The validation report states zero failures for the measured paths. That statement should be interpreted honestly. It means that the benchmarked workflow behaved as expected during the measured run. It does not mean that every possible failure mode, attack, physical compromise, manufacturing condition, network condition, or deployment scenario has been eliminated. Strong public communication should avoid overclaiming. Credibility increases when PahiLabs explains exactly what was measured and what the result means.

For clients, zero failures in this context means the tested devices repeatedly authenticated, securely ingested data, handled a rehandshake, and maintained stable trust behavior in the measured environment. That is meaningful because trust products need repeatability. A single successful operation can be a demonstration. One hundred repeated operations per device are stronger evidence that the workflow is under control. When that result is paired with explicit rejection cases, the benchmark becomes more persuasive.

The most useful phrase for client communication is this: the measured path behaved consistently under repeated use. That statement is specific, defensible, and useful. It tells a buyer that the technology has moved beyond a concept and into a measurable workflow. It also leaves room for normal pilot evaluation, security review, and deployment-specific testing.

Why clone and tamper rejection matter

The benchmark package reports two rejection outcomes that should be emphasized: a cloned ESP32-C3 image was rejected due to hardware identity mismatch, and a tampered ESP32-C3 runtime was rejected due to attestation mismatch. These are public-safe summaries of security behavior. They do not reveal how the underlying checks are implemented, but they communicate the buyer-relevant outcome clearly.

Clone resistance matters because device copying is a practical threat. In many IoT environments, a device image can be copied during development, repair, or compromise. If copied state alone is enough to impersonate an enrolled device, the security model is weak. The reported clone rejection helps PahiLabs explain that MAIA-IOT is not merely checking whether a message has the right shape. It is making a trust decision about the device relationship.

Runtime tamper rejection matters because device behavior can be modified while the device still appears operational. A simple connectivity test may pass even when local code has changed. A trust product needs a way to treat unexpected runtime state as a risk signal. The benchmark result gives PahiLabs a concise and defensible claim: in the measured validation package, ordinary tampering of the tested runtime was not silently accepted.

What clients can trust at the evaluation stage

A client evaluating MAIA-IOT should trust the technology for the right reasons. They should trust that PahiLabs has a concrete validation package, that the product has been exercised on named device classes, that the benchmark includes repeated operations, that the server behavior is explicit, and that rejection scenarios are part of the evidence story. They should not be asked to trust vague language such as military-grade security or unbreakable authentication. Those phrases are not useful in serious enterprise conversations.

The right trust posture is pilot-ready, evidence-backed, and reviewable. Pilot-ready means the technology has a practical story for devices, gateways, and backend ingest. Evidence-backed means the claims can be connected to measured behavior. Reviewable means PahiLabs can explain what was tested, what was not tested, what assumptions exist, and what a client should validate in their own environment.

This posture is stronger than overclaiming. Buyers do not expect a device trust layer to solve every deployment risk on day one. They expect honest evidence, clear integration paths, and a plan for evaluation. MAIA-IOT can meet that expectation by showing the benchmark summary, explaining the device classes, describing the metrics, and offering a pilot that measures the client's own workflow.

Integration implications for pilots

The benchmark evidence supports a clear pilot structure. A client can begin by selecting one device class, one gateway or server environment, and one data flow that matters. PahiLabs can then integrate the device-side and verification-side components around that flow, measure repeated authentication and secure ingest, and document acceptance and rejection outcomes. This keeps the pilot practical. It avoids a broad platform rollout before the buyer has evidence in their own environment.

The MAIA-IOT integration model should be described as device, gateway, and ingest integration. The device or gateway produces protected traffic. The verification side decides whether that traffic should enter the client's system. The client's existing applications, dashboards, databases, and operational tools continue to handle business logic after data is accepted. That separation is useful because it means MAIA-IOT can be evaluated as a trust layer rather than a full replacement for the client's architecture.

A strong pilot should define success metrics before work begins. Useful metrics include valid operation completion, secure ingest completion, rehandshake behavior, rejection of copied state, rejection of modified runtime behavior, observed throughput, operator experience, and deployment complexity. The public benchmark package provides a template for that conversation.

Questions clients should ask during evaluation

A serious client should use the MAIA-IOT benchmark as the start of an evaluation conversation, not the end of one. The first question is which device class matters most for the buyer's environment. A telecom buyer may care about gateways and customer-premises equipment. An industrial buyer may care about sensor controllers and edge nodes. A product manufacturer may care about boards used in a first-generation device line. The benchmark shows that PahiLabs has tested multiple board classes, but the client's own pilot should select the hardware that best reflects the deployment decision.

The second question is which data flow should be protected first. Device security becomes easier to evaluate when the pilot is tied to one concrete data path. For example, a client might protect status telemetry, maintenance events, command acknowledgements, or exception alerts. The protected path should be important enough to prove value but narrow enough to evaluate cleanly. This makes the benchmark repeatable inside the client's environment and prevents the pilot from becoming a vague platform exploration.

The third question is what rejection behavior must be demonstrated. A buyer should not only ask whether valid devices can send data. They should ask what happens when device state is copied, traffic is repeated, runtime behavior changes, or a gateway receives traffic from an unexpected source. The public MAIA-IOT benchmark already emphasizes rejection of clone and tamper

scenarios. A client pilot can extend that idea by defining the rejection cases that matter most to the target deployment.

How this white paper should be used by buyers

This web white paper is designed for early trust formation. A business reader can use it to understand why MAIA-IOT deserves attention. A technical reader can use it to prepare questions for an architecture briefing. A procurement or risk reviewer can use it to identify what evidence should be requested next. The document is not a replacement for a full security review, source review, deployment guide, or contractual assurance package. It is a structured public entry point.

In practice, the best next step is a scoped evaluation call. During that call, PahiLabs should map the client's device environment, target hardware, gateway or server path, expected throughput, security concerns, and pilot success criteria. The benchmark can then be used as a reference model for what the client should expect to see: repeated valid operation, clear trust-state handling, measured runtime behavior, and explicit rejection of invalid device conditions.

Limitations and responsible claims

Responsible communication is part of trust. The website should not claim that MAIA-IOT eliminates all IoT risk. It should not claim universal throughput. It should not imply that a benchmark run is the same thing as certification, third-party penetration testing, or production assurance across every hostile physical scenario. Instead, it should say that the validation package demonstrates repeated live operation and explicit rejection behavior in the measured environment.

For hostile physical access, production deployments often require additional hardware protections, secure boot configuration, key protection, supply-chain controls, update governance, and operational monitoring. These points do not weaken the MAIA-IOT message. They make the message more credible. Serious buyers know that device security is layered. A product that can explain its role in that layered model earns more trust than a product that pretends one control solves everything.

The benchmark is therefore best used as evaluation evidence. It helps a buyer decide whether MAIA-IOT deserves a pilot. It helps technical reviewers ask better questions. It helps investors understand that the product has measurable behavior. It helps sales teams talk about device trust without exposing internal protocol mechanics.

Conclusion

MAIA-IOT should be trusted because it is presented with evidence, not because it asks for belief. The validation package names the device classes, reports repeated live authentication and secure ingest, includes a rehandshake event, summarizes throughput, and documents clone and tamper rejection outcomes. That is the right kind of evidence for an early enterprise evaluation.

The benchmark does not need to disclose intellectual property to be useful. Buyers do not need the protocol internals on a public website. They need to understand what was tested, why the result matters, how the technology integrates, and how the result can be repeated in a pilot. This white paper gives them that structure.

The practical takeaway is simple: MAIA-IOT turns device communication into a measured trust workflow. The right devices were accepted repeatedly, wrong device or runtime conditions were rejected in the validation package, and the

result can be used as the basis for a scoped pilot. That is a credible, client-ready trust message.

PahiLabs

Innovating Security & Intelligence for a Connected World

Quick Links

Home

About Us

Products

Consultancy & Services

Resources & Blog

Contact & Support

Products

MAIA SSO

MAIA-IOT

MAIA-PQ

LENS

Tok2DBs

Contact Us

support@pahilabs.com
info@pahilabs.com

IPN - Building C,
Rua Pedro Nunes, 3030-199, parish of Santo António dos Olivais,
municipality of Coimbra, Portugal

Supported by

1 2 9 0
UNIVERSIDADE D COIMBRA

BUSINESS