CAPSEAL TECHNICAL WHITEPAPER · TW-01

Provenance-Gated Evidence Capture: Hardware-Rooted Attestation and Multi-Layer Synthetic-Media Detection in the CapSeal Architecture

A formal treatment of the CapSeal trust pipeline: the device-resident attestation and sealing protocol, the verification-time detection stack, the fusion calculus that produces a payment-grade verdict, and the security arguments that bound an adversary's forgery advantage.

SYPTIME PTY LTD · APPLIED CRYPTOGRAPHY & MEDIA FORENSICS GROUP

VERSION 1.0 · AUGUST 2026 · CLASSIFICATION: PUBLIC (PARAMETERS WITHHELD)

DOCUMENT ID: CS-TW-01-2026

ABSTRACT

Generative image models have collapsed the marginal cost of fabricating photographic evidence to near zero, while post-hoc forensic detection degrades with every model generation: detection is a moving-target problem with an adversary that retrains faster than defenders can re-annotate. CapSeal inverts the problem. Rather than asking whether an artifact *looks* synthetic, CapSeal requires every artifact to *prove its own origin* at capture time, using signing keys resident in the device's hardware security module, a capture-session attestation protocol with server-supplied freshness, a sensor-path integrity battery that binds photons to silicon, and a sealed provenance manifest whose verification reduces to standard cryptographic assumptions. A verification-time detection stack - cryptographic, attestational, physical-statistical, and generative-artifact layers fused through calibrated likelihood ratios with a conformal abstention region - converts the seal (or its absence) into one of three operational verdicts. We give a formal adversary model with six capability classes, specify the attestation and sealing protocols, derive the forgery-advantage bound (Theorem 1), analyse injection and analog-hole attacks (Theorems 2-3), and characterise the detection layers with their underlying estimators: PRNU correlation with peak-to-correlation-energy statistics, CFA

periodicity via expectation-maximisation, double-quantisation ghost analysis, azimuthally averaged spectral discrepancy for upsampling artifacts, and cross-modal ego-motion coherence. Concrete model parameterisations, decision thresholds, hardening schedules and red-team corpora are proprietary and withheld; the architecture is published, the instantiation is not.

1 Introduction

Institutional decision systems - claims settlement, credit origination, program administration - were engineered on an assumption so deeply embedded it was rarely stated: that producing a convincing photograph or document of an event that did not occur is costly. Diffusion-based image synthesis and instruction-tuned editing models falsified the assumption. The relevant question is no longer whether a submitted artifact appears authentic under inspection, because appearance is precisely the property generative models optimise. The relevant question is whether the artifact can demonstrate a chain of custody from photons striking a physical sensor to bytes entering an adjudication workflow.

Two research programmes address the crisis from opposite ends. *Passive forensics* analyses statistical residue of manipulation in the artifact itself - sensor noise, interpolation traces, compression history, physical inconsistency [1, 4, 5, 6, 8]. It requires no cooperation at capture, but it fights an adversary who controls the generator and can regularise away each published cue; empirically, detector accuracy decays sharply across generator generations [7]. *Active provenance* binds authenticity metadata to the artifact at creation. Its guarantees are cryptographic rather than statistical, but naive implementations are vacuous: a signature only proves that *someone with a key* signed *some bytes*. If the key is extractable, the signing environment forgeable, or the input to the camera stack injectable, provenance signs the fraud.

CapSeal is an integrated system built on the position that neither programme suffices alone. Provenance without capture-path integrity signs synthetic frames; forensics without provenance loses the arms race. CapSeal therefore (i) roots signing keys in device hardware security modules and verifies that rooting via platform key-attestation certificate chains; (ii) executes a challenge-response capture-session protocol that binds each seal to server freshness, precluding replay and pre-computation; (iii) runs an on-device sensor-path integrity battery designed to ensure the signed frame originated at the physical camera during the attested session; (iv) seals a structured provenance manifest under a Merkle commitment permitting redacted disclosure; and (v) verifies every submission through a layered detection stack whose outputs fuse into a calibrated three-way verdict: `PASS`, `REVIEW`, `HOLD`.

This paper specifies the architecture at a level sufficient for security evaluation and insufficient for reproduction. Sections 2-3 formalise the threat model and security goals. Sections 4-6 specify the attestation root, the capture-session protocol, and the sensor-path battery. Section 7 specifies sealing and the transparency log. Section 8 develops the verification-time detection stack and its fusion calculus. Section 9 states the security theorems and analyses residual risk. Sections 10-11 treat privacy and performance. Throughout, quantities marked **[withheld]** denote proprietary parameterisations.

2 Adversary Model

Let $\mathscr{A}$ denote an adversary attempting to cause the CapSeal verifier to emit `PASS` on evidence of an event that did not occur as represented. We stratify $\mathscr{A}$ into capability classes ordered by cost:

CLASS	CAPABILITIES	REPRESENTATIVE ATTACK	EST. MARGINAL COST
`A0`	Consumer editing / generative apps; no system access	Edit damage photo after capture; submit via app	< $10, minutes
`A1`	OS-level tampering on own device: root/jailbreak, hooking frameworks	Hook camera callbacks; feed stored image to SDK	$\$10^1$-$10^2$, hours
`A2`	Virtual camera / display-mediated injection ("analog hole")	Photograph a high-fidelity screen or print of a synthetic scene	$\$10^2$-$10^3$
`A3`	Emulation and attestation-spoofing infrastructure; farmed devices	Custom ROM emulating attestation responses; device farms replaying stale verdicts	$\$10^3$-$10^4$
`A4`	Physical staging of real scenes; collusive insiders	Genuinely photograph pre-staged or borrowed damage	$\$10^3$-$10^5$
`A5`	Hardware-invasive key extraction; zero-day TEE compromise	Decapsulate SE/StrongBox; exfiltrate device key	> $\$10^5$, lab-grade

The design target is stated economically rather than absolutely: CapSeal must raise the marginal cost of a `PASS`-grade forgery above the expected fraud proceeds for classes A0-A3, force class A4 into behaviours detectable by longitudinal and cross-carrier analytics (Section 8.6), and ensure class A5 attacks are per-device, non-scalable, and revocable. Formally:

> **Definition 1 (Seal-forgery game $\mathsf{Forge}_{\mathscr{A}}$).** The challenger provisions an honest device D with enrolment as in Section 5. $\mathscr{A}$ receives oracle access to (i) honest capture sessions on D, (ii) the public verification interface, and (iii) enrolment of adversarial devices under $\mathscr{A}$'s control. $\mathscr{A}$ outputs a tuple (m, σ, τ) - manifest, seal, session transcript. $\mathscr{A}$ wins if `Verify`(m, σ, τ) = `PASS` and m attests a capture event that did not occur on an honest enrolled device within the freshness window. $\mathrm{Adv}^{\mathrm{forge}}_{\mathscr{A}}(\lambda) = \Pr[\mathscr{A} \text{ wins}]$.

Security goals, each mapped to mechanisms in later sections: **G1 unforgeability** (no valid seal without an honest hardware key and live session; Thm. 1); **G2 injection resistance** (signed frames originate at the physical sensor; Sections 6, 9.2); **G3 freshness** (no replay or pre-computation across sessions; Section 5.3); **G4 spatiotemporal binding** (device, time, place bound under the seal; Section 7); **G5 tamper evidence** (any post-capture modification invalidates verification; Section 7); **G6 graceful degradation** (absence of a seal routes to human review, never auto-rejection; Section 8.5); **G7 auditability** (verdicts replayable offline years later; Section 7.4).

3 Notation and Assumptions

λ denotes the security parameter. H is a collision-resistant hash (SHA-256); signatures are ECDSA over P-256 as exposed by platform secure elements, with EUF-CMA advantage $\mathrm{Adv}^{\mathrm{euf}}_{\mathrm{ECDSA}}(\lambda)$. $\varepsilon_{\mathrm{SE}}$ denotes the probability of secure-element key extraction or signing-oracle abuse outside the attested code path within the deployment horizon - an assumption on Apple Secure Enclave / Android StrongBox tamper resistance, priced per class A5. $\varepsilon_{\mathrm{att}}$ bounds acceptance of a forged platform attestation chain (compromise of Apple/Google attestation CAs or verifier misvalidation). ε_{sensor} bounds the sensor-path battery's false-accept rate against injection (Section 6); it is empirical, adversarially estimated, and re-estimated per hardening cycle **[withheld]**. q_s is the number of capture sessions an adversary may observe or initiate.

4 Architecture Overview

CapSeal comprises five planes. The **capture plane** (SDK, on-device) mediates the camera session and computes frame commitments in real time. The **attestation plane** (on-device HSM + platform services + CapSeal attestation verifier) proves device, application, and session integrity. The **sealing plane** constructs and signs the provenance manifest and anchors it to the transparency log. The **verification plane** executes the layered detection stack. The **decision plane** fuses layer outputs into the operational verdict and emits the replayable proof object. Planes are separated by narrow, versioned interfaces; compromise of any single server-side plane cannot mint seals, because seal issuance requires a signature under a key that never leaves device hardware.

Figure 1. The five CapSeal planes. Seal issuance requires a hardware-resident key; no server-side compromise can mint seals.

5 Hardware-Rooted Attestation

5.1 Roots of trust

CapSeal treats the device's isolated security processor as the root of trust: the Secure Enclave Processor on iOS-class devices and the StrongBox Keymaster (discrete secure element) or TEE-backed Keymaster on Android-class devices. Two properties are load-bearing. First, *non-exportability*: keys generated inside the module can sign but never leave; extraction requires class-A5 invasive attack, priced into ε_{SE}. Second, *platform key attestation*: the module emits an X.509 chain, rooted in the platform vendor's attestation CA, whose leaf certificate embeds a structured description of the key and its environment - on Android, the `KeyDescription` ASN.1 sequence carrying `attestationSecurityLevel`, verified-boot state (`deviceLocked`, `verifiedBootState`, vbmeta digest), OS version and patch level, and the binding of the key to the calling application's package and signing-certificate digest; on iOS, the App Attest attestation object binding the key to the team and bundle identifiers with a receipt for ongoing assessment [11, 12].

The CapSeal attestation verifier validates: chain of trust to pinned vendor roots; certificate transparency of intermediates; revocation status (vendor CRL/OCSP feeds, with fail-closed semantics for StrongBox-class claims); security-level admissibility (policy lattice over {StrongBox, TEE, software} × boot states, where software-level keys are inadmissible for sealing); application binding equal to the registered CapSeal SDK embedding; and patch-level recency against a rolling admissibility window **[withheld]**. Devices failing admissibility do not fail capture - they capture unsealed, and the submission routes as `UNVERIFIED` (goal G6).

5.2 Enrolment ceremony and key hierarchy

At first run, the SDK executes enrolment: (1) generate P-256 keypair (pk_D, sk_D) inside the security module with attestation challenge c_e supplied by the CapSeal enrolment service; (2) submit the attestation chain; (3) the verifier performs the Section 5.1 checks and, on success, issues a device credential $cred_D = Sig_{CS}(pk_D \parallel \text{policy-class} \parallel \text{epoch})$ recorded against a monotonically increasing enrolment counter. Re-enrolment on the same hardware is bound to the platform's key-rotation and app-integrity signals to resist device-farm churn; enrolment velocity anomalies feed the risk lattice (Section 8.2). Per-session keys are not used for sealing; instead, session binding is achieved in the protocol transcript (Section 5.3), keeping the signing key singular, attested, and revocable. Revocation is a first-class operation: compromise indicators (Section 9.3) revoke $cred_D$, and all seals chain-verify against credential validity at their anchored timestamp - revocation is forward-effective, past seals remain verifiable.

5.3 Capture-session attestation protocol

Sealing occurs only inside an attested session. The protocol (Figure 2) binds server freshness, application integrity, the sensor-path battery verdict, and the frame commitments into one transcript signed inside the security module.

```
(1)  D -> S :  hello, cred_D, caps                        # capability & policy negotiation
(2)  S -> D :  n_s, t_s, pol                          # 128-bit nonce, server time, session policy
(3)  D      :  open camera session; battery B starts        # Section 6, continuous
(4)  D      :  for each candidate frame f_i:
                 h_i = H(tile-Merkle-root(f_i))            # streaming commitment, pre-encode
(5)  D      :  m = manifest(h_sel, meta, geo, t_d, B.summary)
(6)  D      :  tr = H(n_s || t_s || m || app_claim)
(7)  D -> S :  m, sigma = Sign_skD(tr), att_refresh         # signature computed in SE/StrongBox
(8)  S      :  verify att_refresh, cred_D, sigma, |t_s - t_d| < delta,
               n_s unused, B.summary within pol            # delta: skew window [withheld]
(9)  S      :  anchor(H(m || sigma)) -> transparency log; issue proof object P
```

Figure 2. Capture-session protocol. The nonce n_s precludes pre-computation; the signature is computed inside the hardware module over a transcript that includes the sensor-battery summary, so a frame that failed the battery cannot be retroactively blessed.

Three consequences merit emphasis. **(i) No offline forgery.** Because n_s enters the signed transcript, an adversary cannot pre-compute seals for a library of synthetic images; every seal is bound to a session the server issued in real time. Deferred-connectivity capture is supported by a bounded-lifetime device-local session ticket with a hardware monotonic counter, reconciled at next contact; the freshness window widens and the risk lattice weights the seal accordingly. **(ii) Commit-before-encode.** Frame commitments are computed on sensor-delivered buffers ahead of user-space encoding, narrowing the window in which a hooked encoder could substitute content; buffer-provenance checks in the battery (Section 6.1) police that window. **(iii) Battery-in-transcript.** The integrity battery's summary vector is inside the signed transcript. The server does not trust the device's battery verdict - it re-scores the raw battery features under server-held models - but the transcript binding prevents a compromised app from replacing the vector post-signature.

6 Sensor-Path Integrity Battery

Attestation proves *which silicon signed*; it does not by itself prove *what the camera saw*. The battery B is a concurrent suite of tests, executed during the live session, designed to bind the committed frames to photons on the physical sensor. Tests fall into four families; each emits a feature vector, not a binary, and the family outputs are fused server-side (Section 8.4). Per-test thresholds, model weights and the test-scheduling policy are **[withheld]**.

6.1 Pipeline provenance

The SDK validates that frame buffers originate from the platform camera HAL rather than a virtual camera, verifying provider identity and characteristics against the attested device model's sensor inventory; cross-checks buffer timestamps against the sensor's declared frame-interval jitter distribution; and inspects the process for instrumentation artifacts associated with hooking frameworks (in-memory code-integrity probes, PLT/GOT anomaly scans, debugger and emulator indicia). On admissible hardware these checks corroborate, and are corroborated by, the boot-state claims in the attestation chain: a rooted device asserting a locked verified-boot state is a contradiction resolved in favour of the chain.

6.2 Display-mediated capture (screen-of-screen)

Class-A2 adversaries photograph a display or print of a synthetic scene: the photons are real, the scene is not. Countermeasures exploit physical signatures of emissive and printed surfaces: (a) *spectral moiré* - the beat frequencies between the display's subpixel lattice and the camera's colour-filter array produce characteristic peaks in the 2-D spectrum whose loci depend on the pixel-pitch ratio; the detector scans the admissible loci manifold rather than fixed frequencies, resisting distance/angle evasion; (b) *temporal luminance banding* from display refresh and PWM dimming interacting with rolling-shutter row sampling, modelled against the attested sensor's row-readout time; (c) *specular and planarity geometry* - multi-frame parallax during natural handheld motion estimates scene depth structure; a planar depth field with display-glass specular residue at plausible scale is strong A2 evidence; on depth-capable devices, the ToF/LiDAR channel is cross-checked directly; (d) *dynamic-range and gamut envelopes* of emissive surfaces versus natural scenes under the reported exposure triple. Family (c) generalises to printed media, which defeat (a)-(b) but not planarity.

6.3 Cross-modal ego-motion coherence

During the session the SDK samples the IMU at high rate and computes sparse optical flow on preview frames. Let ω(t) be gyroscope angular velocity and v(t) the flow-derived camera rotation estimate. For genuine handheld capture, v is a filtered, latency-shifted transform of ω; the battery estimates the latency and computes windowed coherence

$$C = max_{\tau\in[0,\tau_{max}]}\, corr(\ v(t), (K \ast \omega)(t-\tau)\ ) \tag{1}$$

with kernel K fitted per device class. Injected video streams exhibit motion decoupled from the device's inertial state ($C \approx 0$), and replayed pans exhibit coherent but distribution-atypical trajectories captured by a sequence model over (ω, v) **[withheld]**. A liveness micro-interaction (guided reframe) can be demanded by session policy pol when the risk lattice requests step-up, injecting entropy an injection rig must track in real time across both modalities simultaneously.

6.4 Sensor physics

Rolling-shutter geometry (row-time consistency of moving edges against the attested readout rate), photon-transfer noise (variance-vs-signal linearity consistent with the sensor's conversion gain), lens-shading and chromatic-aberration radial fields consistent with the attested optics, and auto-exposure/AWB convergence dynamics matching the platform ISP's control-loop signature. Each is a weak signal; jointly, they force an injection rig to emulate an entire imaging physics stack, per device model, in real time.

7 Sealing, Manifests, and the Transparency Log

7.1 Manifest structure

The manifest m is a canonical CBOR structure of assertions: content commitment (Merkle root over 256×256 tiles under H, enabling redacted disclosure - Section 10); capture parameters (exposure triple, focal metadata, sensor identifier class); attestation claim digest; battery summary vector; spatiotemporal claim (t_d from the module's secure clock where available, geodetic position generalised to a policy-tiered precision); and lineage (SDK version, policy id, session id). Assertions are individually hashed into an assertion tree whose root is the signed object, so any assertion may later be disclosed or withheld without re-signing.

7.2 Signature suite and time

σ = ES256 under sk_D, verified against $cred_D$ and the pinned vendor chain. Server-side, the anchored digest receives an RFC 3161 timestamp from an independent TSA [13], decoupling long-term temporal proof from CapSeal's own clock. Algorithm agility is versioned in the manifest; migration to post-quantum signatures is a policy event, not a format break.

7.3 Redactable disclosure

Because content is committed tile-wise, a claimant's number plate or bystander's face can be redacted in the disclosed artifact while the seal still verifies over the undisclosed tiles' hashes: verification confirms that *nothing outside the declared redaction set changed*. Redaction sets are themselves declared assertions, so silent redaction is impossible.

7.4 Transparency log

Every issued proof object's digest is appended to a Merkle transparency log in the style of Certificate Transparency [14]: signed tree heads, inclusion proofs delivered with the proof object, and consistency proofs available to auditors, making retroactive seal insertion or deletion publicly detectable. Verification is therefore possible offline (chain + signature + inclusion proof) and remains sound if CapSeal-the-company disappears; issuance, by contrast, is impossible without a hardware key enrolled under the live protocol. This is the precise technical sense of the product claim *verifiable by anyone, producible only by CapSeal*.

8 The Verification-Time Detection Stack

Verification is layered so that each layer conditions the interpretation of the next. Cryptography is dispositive where it applies; forensics carries the residual mass, principally for unsealed submissions (legacy channels, non-app intake) and for semantic fraud on genuine captures.

8.1 L0 - Cryptographic verification

Chain validation, signature verification, transcript freshness, inclusion proof, revocation state at anchored time. L0 outcomes are trichotomous: `VALID`, `INVALID` (a seal that fails - the single strongest fraud signal in the system, since honest pipelines do not corrupt seals), and `ABSENT`.

8.2 L1 - Attestation policy evaluation

The risk lattice $\mathcal{L}$ maps the attestation tuple (security level, boot state, patch recency, enrolment history, session freshness class, battery summary) to a prior log-odds contribution ℓ_1. $\mathcal{L}$ is a partial order, not a score: e.g. (StrongBox, locked, fresh) dominates (TEE, locked, stale) which is incomparable with (StrongBox, locked, deferred-ticket). Lattice calibration is performed against longitudinal outcome data and is **[withheld]**.

8.3 L2 - Physical-statistical forensics

Applied to sealed content as a defence-in-depth cross-check and to unsealed content as primary evidence. Principal estimators:

- **PRNU sensor fingerprinting.** The photo-response non-uniformity pattern K of a sensor is a multiplicative noise fingerprint [1]. With noise residual $W = I - F(I)$ for a denoising filter F, the detector evaluates the normalised correlation $\rho = \mathrm{corr}(W, \hat{I}\cdot\hat{K})$ and its peak-to-correlation-energy ratio, $\mathrm{PCE} = \rho^2_{peak} / (mn)^{-1}\Sigma_{(u,v)\notin\mathcal{N}}\, \rho^2(u,v)$, against per-device-class acceptance regions. For enrolled devices CapSeal maintains an encrypted fingerprint estimate updated across sessions; a sealed frame whose PRNU rejects the enrolled sensor is treated as an inconsistency of the highest severity.
- **CFA/demosaicing periodicity.** Bayer interpolation induces a periodic linear dependency among neighbouring samples; the EM estimator of [4] recovers the probability map of interpolated pixels, and locally aperiodic regions localise splices and inpainting.
- **Compression history.** Double-quantisation ghosts and DCT-coefficient distribution tests [5] expose re-encoding inconsistent with the manifest's declared single-pass pipeline; block-grid misalignment localises composited regions.
- **Physical light and optics.** Low-order spherical-harmonic illumination estimation across objects and faces [6], cast-shadow geometric consistency, and radial chromatic-aberration displacement fields inconsistent with the attested lens model.

8.4 L3 - Generative-artifact detection

An ensemble targeting synthesis processes rather than semantic content: azimuthally averaged power-spectrum discrepancies and upsampling harmonics characteristic of decoder topologies [8]; learned patch-level detectors trained with the augmentation regime of [7] for cross-generator generalisation; reconstruction-residual tests that measure how cheaply a candidate image is re-explained by a bank of latent-generative priors; and colour-space co-occurrence features robust to recompression. Ensemble composition, training corpora, and refresh cadence are [withheld]; the operational stance assumes every individual detector decays and engineers the ensemble for graceful decay under generator drift, with drift monitors on score distributions triggering re-training (Section 8.6).

8.5 Fusion and the verdict calculus

Each layer i emits a calibrated log-likelihood ratio $\ell_i = \log [\, p(x_i \mid \text{fraud}) / p(x_i \mid \text{genuine}) \,]$, calibrated per channel class by isotonic regression on held-out adjudicated outcomes. The fused statistic is

$$\Lambda = \Sigma_i\, w_i\, \ell_i, \quad w \in \mathscr{W}_{pol} \tag{2}$$

with channel-conditional weights (a sealed submission concentrates weight on L0/L1; unsealed on L2/L3). Verdicts are produced by a conformal gate: thresholds (t_{lo}, t_{hi}) are set to guarantee, with finite-sample validity under exchangeability, a bounded false-`HOLD` rate on genuine claims at target coverage; $\Lambda < t_{lo} \Rightarrow$ `PASS`, $\Lambda > t_{hi} \Rightarrow$ `HOLD`, else `REVIEW`. The `REVIEW` region is a designed abstention class, not indecision: it is the formal implementation of goal G6, and its width is a policy parameter priced against reviewer capacity. Operating points, coverage targets and the conformal calibration set are [withheld].

8.6 Longitudinal and cross-submission analytics

Class-A4 staging defeats single-image physics by construction; it is attacked longitudinally: PRNU linkage of media across unrelated submissions, near-duplicate retrieval over perceptual and geometric embeddings, geodesic-temporal impossibility tests across a claimant's history, and - where deployed across multiple institutions - cross-carrier re-presentation checks over salted digests, so one loss cannot be sold twice. These analytics consume seal metadata, not disclosed imagery, preserving the privacy posture of Section 10.

9 Security Analysis

9.1 Unforgeability

Theorem 1 (Seal unforgeability). For any probabilistic polynomial-time adversary $\mathscr{A}$ in classes A0-A3 making at most q_s session queries,

$$Adv^{forge}_{\mathscr{A}}(\lambda) \leq Adv^{euf}_{ECDSA}(\lambda) + \varepsilon_{SE} + \varepsilon_{att} + q_s \cdot \varepsilon_{sensor} + q_s^2 \cdot 2^{-|n_S|}.$$

Proof sketch. A winning forgery either (i) produces a valid signature under an honest pk_D without the module - reducing to ECDSA EUF-CMA or SE compromise; (ii) passes verification under an adversarial key whose attestation chain falsely claims admissible hardware - event bounded by ε_{att}; (iii) uses an honest device and live session but signs non-sensor content - requiring the battery to false-accept in that session, bounded per session by ε_{sensor}; or (iv) replays or predicts a nonce - a birthday event over the 128-bit space. The transcript construction (Fig. 2, step 6) makes the cases exhaustive. ∎

The bound makes the engineering priorities explicit: the cryptographic and attestation terms are negligible under standard assumptions; the system's true security budget is spent on ε_{sensor}, which is empirical and adversarial. This is why the battery is an ensemble across independent physical modalities (Section 6): an injection rig must defeat all families simultaneously, in real time, on attested hardware it does not fully control.

9.2 Injection and the analog hole

Theorem 2 (Injection resistance, informal). Under the buffer-provenance and boot-state corroboration of Section 6.1, software-level frame injection on admissible hardware requires a contradiction between the platform attestation chain and runtime state, occurring with probability $\leq \varepsilon_{att} + \varepsilon_{hook}$, where ε_{hook} bounds undetected user-space hooking on a locked, patched device.

Theorem 3 (Analog-hole cost shifting, informal). Display-mediated attacks are information-theoretically undetectable in the limit of a perfect re-imaging channel; the battery's families (a)-(d) instead impose a rig-cost function ascending in display fidelity, optical decoupling, motion tracking and depth spoofing. CapSeal's claim is economic: the cheapest rig passing all families concurrently exceeds the A2 budget envelope, and its residual failure probability against family (c) is bounded by the depth-spoofing term ε_{depth} **[withheld]**.

We regard Theorem 3's honesty as a feature: no capture system can make photographing a screen *logically* impossible. CapSeal's design goal is to make it *economically irrational at scale* and *individually risky*, since every attempt executes inside an attested, transcript-bound session that preserves the forensic record of the attempt.

9.3 Residual risks and compromise response

(i) **A5 key extraction** yields one device's key: seals remain per-device attributable, anomaly detection over per-credential issuance patterns triggers revocation, and no cross-device amplification exists. (ii) **Platform attestation CA compromise** is a platform-ecosystem event; CapSeal fail-closes on vendor advisories and re-scores affected epochs. (iii) **Model-drift decay** in L3 is monitored via score-distribution drift statistics with scheduled ensemble refresh. (iv) **Collusive staging (A4)** is out of scope for capture integrity and in scope for Section 8.6 analytics and downstream intelligence layers. Every residual is mapped to a detection, revocation, or re-scoring pathway; none is silent.

10 Privacy and Compliance Posture

Sealing is data-minimising by construction: verification consumes digests, assertions and proofs, not imagery; geodetic assertions are tiered (exact / suburb / region) under policy; tile-Merkle commitments give cryptographic redaction (Section 7.3); the transparency log stores digests only; and battery feature vectors contain no biometric templates. The architecture maps to Australian Privacy Principles and GDPR data-minimisation and purpose-limitation requirements, with retention governed by the institution's schedule - the proof object remains verifiable regardless of where the disclosed artifact resides.

11 Performance Envelope

STAGE	LOCUS	DESIGN BUDGET
Frame commitment (per frame, tile-Merkle)	on device	≤ 12 ms

STAGE	LOCUS	DESIGN BUDGET
Battery families (concurrent, amortised)	on device	≤ 40 ms/frame budget class
Module signature (ES256)	SE / StrongBox	≤ 80 ms
End-to-end seal, capture to proof object	device + service	p95 ≤ 1.8 s
Server verification (L0-L1)	service	p95 ≤ 120 ms
Full-stack verification incl. L2-L3	service	p95 ≤ 2.5 s

Budgets are design targets for admissible device classes; the SDK degrades battery family depth, never seal correctness, on constrained hardware.

12 Conclusion

CapSeal replaces an unwinnable classification race with a provable custody question. The attestation plane roots trust in vendor silicon and verified boot; the session protocol binds every seal to live server entropy; the sensor battery binds signed bytes to physical photons across independent physical modalities; the sealing plane produces redactable, transparency-logged, offline-verifiable proofs; and the detection stack converts cryptographic certainty and forensic evidence into a calibrated, abstention-aware operational verdict. The forgery bound of Theorem 1 localises the entire security budget into one adversarially estimated quantity, ε_{sensor}, and the engineering programme of Sections 6 and 8 is precisely the continuous minimisation of that quantity against a live red-team loop. The architecture is published here for evaluation. The parameterisation that makes it bite - thresholds, lattices, ensembles, corpora, schedules - is the product, and it stays withheld.

References

[1] J. Lukas, J. Fridrich, M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Trans. Information Forensics and Security*, 1(2), 2006.

[2] M. Goljan, J. Fridrich, T. Filler. Large scale test of sensor fingerprint camera identification. *Proc. SPIE Media Forensics and Security*, 2009.

[3] H. Farid. *Photo Forensics*. MIT Press, 2016.

[4] A. C. Popescu, H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Trans. Signal Processing*, 53(10), 2005.

[5] H. Farid. Exposing digital forgeries from JPEG ghosts. *IEEE Trans. Information Forensics and Security*, 4(1), 2009.

[6] M. K. Johnson, H. Farid. Exposing digital forgeries in complex lighting environments. *IEEE Trans. Information Forensics and Security*, 2(3), 2007.

[7] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, A. A. Efros. CNN-generated images are surprisingly easy to spot... for now. *CVPR*, 2020.

[8] R. Durall, M. Keuper, J. Keuper. Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions. *CVPR*, 2020; J. Frank et al. Leveraging frequency analysis for deep fake image recognition. *ICML*, 2020.

[9] V. Vovk, A. Gammerman, G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005. (Conformal prediction.)

[10] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[11] Android Open Source Project. Verified Boot and Key Attestation documentation; Android Keystore `KeyDescription` attestation schema. developer.android.com, 2025.

[12] Apple Inc. DeviceCheck and App Attest: Establishing your app's integrity. developer.apple.com, 2025.

[13] C. Adams, P. Cain, D. Pinkas, R. Zuccherato. RFC 3161: Time-Stamp Protocol (TSP). IETF, 2001.

[14] B. Laurie, A. Langley, E. Kasper. RFC 6962: Certificate Transparency. IETF, 2013.

[15] IETF RATS Working Group. RFC 9334: Remote ATtestation procedureS architecture. IETF, 2023.

[16] NIST. FIPS 186-5: Digital Signature Standard. 2023.
