

BitTool Signer for Android

Code Security Re-audit Report

Assessment date: 24 October 2026 (Australia/Sydney)
Current baseline: e7e71fbc4b5614b68e993e1eec5dc986b4536a86
Previous baseline: bb1e6567fb2a8cc30dd6e8e9447e69b0f330fa1b
Assessment: AI-assisted code review, fix verification and automated regression checks
Disposition: Three findings closed; three remain open. Production code was not modified.

1. Executive assessment

The latest changes address the three intended defects: partially obscured signing-review touches, cancellation-time cleanup of application-owned secret buffers, and retry of an interrupted static QR scan. Independent checks support closing BTS-01, BTS-03 and BTS-05 within their stated scope.

The engine is unchanged. BTS-02 and BTS-04 remain reproducible against both source and the committed Android bundle. Further testing expands BTS-02 to authority changes and delegation hidden by a dust-payment headline. BTS-06 also includes a bundle test that reports success without loading a bundle when its obsolete resource path is absent.

Release recommendation: hold general production release for valuable wallets. The remaining Solana consent defect and incomplete engine validation are release concerns. Correct the Bitcoin history summary as well, then verify the resulting release build on supported devices. The fixes materially improve the application, but this revision is not yet release-qualified by this assessment.

ID	Status	Severity	Result at the current baseline
BTS-01	Closed	Medium	Strict review controls reject partial-overlay touches on supported APIs
BTS-02	Open	Medium	Mixed Solana summaries omit destruction or changes of authority
BTS-03	Closed	Low	Secret buffers survive the dispatcher return boundary long enough to be wiped
BTS-04	Open	Low	Bitcoin burn plus change still produces a false all-funds-return summary
BTS-05	Closed	Functional	An identical QR callback now resumes interrupted queued work
BTS-06	Open	Validation	Obsolete resource paths cause one failure and one false-green bundle test

Closed severities describe the original findings. No critical/high-severity vulnerability, password bypass, unauthorized signing path or direct private-key extraction path was confirmed. Absence of such a finding is not a guarantee of absence. This is an engineering review of a specific snapshot, not independent certification.

2. Scope and wallet security posture

The current commit was created at 11:01:37 +11:00 on 24 October 2026. No Git remote is configured, so “latest” means the current local HEAD. The tracked working tree was clean. The change from the previous assessment contains nine files: four production files and five test/support files, with 132 insertions and 23 deletions.

All changed production code and its critical callers were reviewed. The work followed secret ownership across suspension, review-to-authorization transitions, and scanner queue/lifecycle behavior. Engine findings were rerun and expanded. The full configured JVM/Android unit-test task set, Node engine suite, release permission/component checks and both release lint variants were invoked. Gradle reused valid incremental/cached results; this was not a clean-room rebuild of every test.

This is a differential re-audit plus broad regression verification. The prior assessment covered wallet storage, cryptography, signing plans, chain parsers, QR/file inputs, the native host and platform configuration. Findings about unchanged modules are carried forward, not presented as a second uniform line-by-line review of the entire repository.

Secrets at rest. The unchanged vault uses a password layer with PBKDF2-HMAC-SHA256 at 600,000 iterations and AES-256-GCM, plus a separate per-wallet Android Keystore AES-GCM device layer. Authenticated metadata and bounded parsing protect wallet identity and expensive decryption work. Password retry records are authenticated and corrupt records fail closed.

Signing authority. Secret-bearing operations require the wallet password. App Lock biometrics do not substitute for transaction authorization. Signing rederives wallet identity/public information from the unlocked secret and reconstructs the approved signing task. Session epochs prevent stale work from publishing signatures after cancellation or request replacement. The new cleanup guards preserve these checks.

Offline boundaries. The release configuration excludes network permissions and wallet backup/device-transfer domains. Release guard tasks enforce permission and component allowlists. The JavaScript engine is hash-pinned and runs in constrained, fresh QuickJS contexts without network/file bridges or runtime eval. The current source rebuild matches the committed bundle byte for byte.

Remaining assurance boundaries. These controls were evaluated through code, host tests and synthetic fixtures. No malicious-overlay device exercise, OEM Keystore qualification, ARM JNI instrumentation, live-chain execution, side-channel assessment or final store-distributed artifact qualification was performed. Managed/native/JavaScript memory erasure remains best effort beyond explicitly owned mutable buffers. Android 8-9 cannot supply the API 29 partial-obscurity flag; their support policy still needs an explicit device-security decision.

3. Open finding: Solana consent summaries

BTS-02 - Material effects are omitted when mixed with a small transfer

Priority / severity: P2 / Medium. **Confidence:** Reproduced in current source and shipped bundle.

Locations: engine/src/solana.js:153-167 (Metaplex burns), :279 (System Assign), :293 (SPL approval), :329 (SetAuthority), and :394-425 (aggregation).

The headline and persisted summary favor entries in the movements collection. Several destructive or authority-changing instructions only contribute fields and warnings. Adding one ordinary transfer can therefore make the principal display look like a payment while omitting the material non-transfer effect.

Current source and bundled-engine reproductions confirm that Metaplex BurnV1 or BurnNft combined with a one-lamport SOL transfer produces a summary of 0.000000001 SOL. The main display shows that amount and the dust recipient. The destruction is absent from the main headline and persisted summary.

Expanded bundled tests reproduce the same understatement for SPL SetAuthority changing AccountOwner, revocation of MintAuthority, System Assign changing an account's owner program, and SPL Approve assigning a delegate. Each mixed transaction again summarizes only 0.000000001 SOL. These are additional manifestations of the existing aggregation defect, not four separately counted vulnerabilities.

Impact and prerequisites. A coordinator or dApp can supply a transaction containing a small payment plus a material destructive/permission operation. The user must still accept the review and provide the wallet password. A misleading primary summary weakens informed consent and leaves an incomplete historical record of what was authorized. This is not a bypass of the password or a mismatch between approved bytes and signed bytes.

Mitigations retained. All six tested Solana combinations remain high risk and retain relevant warnings and instruction details. Metaplex burns are partially decoded; the four authority/delegation combinations are marked fully decoded. Android still requires the applicable risk acknowledgement. These mitigations limit the demonstrated severity; the report does not claim that all evidence of the dangerous operation disappears.

Evidence. evidence/engine-current-and-bundle-repro.mjs and its log record both the display models and independently verified synthetic signatures. The actual committed Android asset was executed in a Node VM; no Android QuickJS execution or chain broadcast is implied.

Required correction. Aggregate material effects, including destruction, delegation, authority transfer/revocation and unknown high-risk instructions. When combined effects cannot be safely reduced to one amount, use an explicit mixed-actions headline and enumerate each relevant asset/account and beneficiary. The main review, signature receipt and history must agree. Preserve warnings and exact signed-byte binding.

Acceptance tests. Cover all six demonstrated combinations, both instruction orders, standalone variants and combinations with multiple transfers. Assert the destructive/permission effect and its target in the primary display and persisted summary. Check source and the generated Android bundle.

4. Other open findings

BTS-04 - Bitcoin burn is excluded from the receipt summary

Priority / severity: P2 / Low. **Confidence:** Reproduced in source and shipped bundle.

Location: `engine/src/resolver.js:161-163`.

The verified-change summary excludes both change outputs and data outputs when calculating sent value. A positive-value OP_RETURN output is a destruction of funds, but when no ordinary payment remains the function says Every output returns to your wallet.

The current reproduction spends 1 BTC: 0.49999 BTC returns as verified change, 0.5 BTC goes to an unspendable OP_RETURN output, and the fee is 0.00001 BTC. The main review correctly displays Burn and a destruction warning. The textual summary incorrectly states that every output returns to the wallet, affecting the receipt and history. The bundled synthetic signature was independently verified.

Required correction. Account for positive-value unspendable outputs separately from payments and change. Preserve explicit destruction in summary/history, including mixed payment/burn/change transactions. Assert conservation of input value across payment, change, burn and fee. This is primarily an inaccurate record problem; the correct pre-sign Burn card is a meaningful mitigation.

BTS-06 - Legacy paths undermine engine validation

Priority / severity: P3 / Validation quality; not a runtime security vulnerability.

Locations: `engine/test/chain-expansion.test.mjs:117,125-126`; `engine/test/engine.test.mjs:131`.

The registry-integrity test still reads absent iOS resource paths. The full engine run reports 210 tests, 209 passes and one failure at this test. A temporary copy changing only the three paths to their Android counterparts passes the targeted registry check: one pass, zero failures, 18 name-filtered skips. Production/test sources were not changed.

An additional gap was confirmed in the browser-bundle test. It attempts to read `BitToolSigner/Resources/signer.js`; a catch immediately returns when the file is absent. Node reports the test as passed, even though the VM initialization and all bundle assertions never run. The legacy file is absent and the Android asset exists at this baseline. Consequently, the 209 reported passes include one vacuous bundle-test pass.

Required correction. Resolve all required artifacts from the Android asset directory and fail when they are missing. Execute isolation/signing assertions against the actual generated bundle. Add a negative check that intentionally removes or misaddresses a required artifact and verifies failure. A green count alone must not hide missing validation inputs.

The independent bundle reproductions in this re-audit provide evidence for the tested cases, but they do not repair or replace the repository's release validation workflow.

5. Verification of the three fixes

BTS-01 - Closed for the original API 29-30 review-touch scenario. `Hardening.kt:54,63-65,84-93` adds a strict modifier that rejects full obscuration and, on API 29+, partial obscuration. The guard consumes the blocked gesture during the initial pointer pass. `SignBar.kt:147,222` applies it to the actual interactive Sign button and risk acknowledgement. Both pinned and scrolling review layouts use the guarded controls; the noninteractive measurement copy cannot authorize.

The updated predicate and Compose gesture tests pass, including blocked partial-overlay touches and ordinary clean taps. Source tracing confirms the actual button modifier reaches its clickable surface. No alternate unguarded entry to signing authorization was identified in the reviewed flow. Password entry retains the ordinary IME-compatible guard. Recovery/delete controls are not newly protected by this change, and API 26-28 cannot expose the partial-obscuration flag. Closure does not claim all sensitive controls or all overlay attack mechanisms are solved. Real API 29/30 and API 31+ device checks remain required.

BTS-03 - Closed for all three reported application-owned buffers.

`DefaultSignerStore.kt:947-958,1065-1078,1084-1096` retains the produced seed/canonical phrase in an outer holder before a dispatcher switch can return. Cleanup is established before suspension; cancellation wipes the holder even when the caller never receives the result. Successful canonical phrases transfer ownership to their caller.

An independent harness explicitly uses separate dispatchers and deterministic cancellation. Public-index derivation wipes the 128-character seed and makes zero engine calls. Signing wipes the seed, closes the mnemonic, clears busy state and publishes no completed task. Canonicalization wipes the captured original character array and closes its `SecretText` holder. All three checks pass. The newly added repository seed-cleanup test was also run against a temporary compilation of the old store and correctly failed its wipe assertion, confirming that it detects the prior defect. Test probes used public fixtures and never printed secret content.

BTS-05 - Closed. `ScannerViewModel.kt:165-168` lets an identical frame already in the queue trigger `drain()` before seen-set deduplication. The processing guard still permits only one in-flight operation. The real camera frame gate reoffers a static code and resets on resume, so the corrected path is reachable.

Four independent checks against the current compiled class pass: static QR resumes after interruption without incrementing the received count twice; queued/in-flight duplicates are not replayed; a queued duplicate resumes the interrupted first item in FIFO order; and a late callback cannot restart a stopped scanner. The existing repository regression also passes. No new confirmed production regression was found in these changes.

6. Automated verification and provenance

Check	Current result and qualification
Android/JVM tests	1,616 tests across 19 tasks; zero failures, errors or skips. Incremental/cached results included. Three tests added since the prior baseline.
Gradle invocation	Successful; 1,185 actionable tasks: 120 executed, 3 from cache, 1,062 up to date. Includes release guard tasks and lint.
Release guards	APK and Play permission/component checks pass; results were up to date for this invocation.
Engine suite	210 tests: 209 reported passes, one failure, zero skips. BTS-06 explains both the failure and one vacuous pass.
Independent cancellation checks	Three of three pass against current code; the new repository test rejects a temporary old-store implementation.
Independent scanner checks	Four of four pass, including real frame-gate retry and queue/lifecycle cases.
Engine reproductions	Both burn defects persist; six Solana mixed-effect cases and the Bitcoin case reproduced in the committed bundle. Synthetic signatures verified.
Registry path diagnosis	Android-path-only temporary adaptation: one targeted pass, zero failures, 18 filtered skips.
Lint	APK and Play release variants each have zero errors and 71 warnings; counts substantially overlap across variants.
Reproducible engine	Rebuilt without writing production assets: byte-identical, 1,181,026 bytes.

The additional lint warning concerns the API 29 partial-obscurity constant. The predicate explicitly gates its use by SDK value, and tests exercise the boundary; no runtime defect was established from that warning. Lint also warns that the plain JVM `core:model` module is not analyzed by Android lint. Its 109 JVM tests are included in the total above.

Committed bundle SHA-256:

```
6933d0d8a6f60faa8222506ac0bd7eef80eb884c9f73e31e9f354b4740f21077
```

Dependencies, native PBKDF2 and engine code did not change. The earlier same-day assessment's 46/46 portable-C PBKDF2 cross-check and npm advisory triage are carried forward; neither is represented as a fresh device test or new advisory query. The earlier npm scan found development-only elliptic/uuid advisories and zero production-only advisories. A fresh current bundle-input check again finds no elliptic, secp256k1, hdkey or uuid modules in the shipped dependency graph. This does not assess every possible supply-chain attack.

7. Release acceptance and evidence

The current revision is safer than the prior baseline. The remaining wallet concern is the accuracy and completeness of transaction consent, especially for Solana ownership/delegation changes. Encryption and successful unit tests cannot compensate for a summary that understates the operation a user is about to authorize.

Before changing this assessment to release-ready:

1. Correct BTS-02, preserving every material destructive/authority effect in the review and persisted summary. Reproduce the six demonstrated combinations against source and the committed Android bundle.
2. Correct BTS-04 and verify burn/payment/change/fee accounting in the receipt and history.
3. Correct BTS-06, require a genuinely green engine suite and demonstrate that missing required artifacts fail validation.
4. Qualify the actual release build on supported devices: malicious-overlay consent checks, secure-window/background behavior, password/IME interaction, Android Keystore lock/recovery behavior, and ARM JNI/QuickJS execution. Define the supported policy for API 26-28.
5. Recheck source-to-bundle pins and release permission/component allowlists after the fixes, and record the exact build/commit tested.

This report and its sibling evidence folder are the deliverables. The earlier `security-audit-2026-10-24` report is preserved as the historical baseline. Its closed findings must not be treated as still open for this revision.

Evidence index. `evidence/README.md` lists commands, harnesses and limitations. `verification-summary.json` contains per-task counts; `baseline.json` records the commit; `reviewed-change.diff` fixes the reviewed scope. Logs preserve the failed engine test, bundle-test false green, independent cancellation/scanner checks, and source/bundle transaction models. SHA256SUMS fingerprints the report and evidence. No real wallet secrets or live transaction broadcasts were used.

Primary technical references. The code-level checks are consistent with [Android tapjacking guidance](#), [Kotlin with Context cancellation semantics](#), and [Android Keystore API documentation](#). References support platform semantics, not a claim that device qualification has been completed.