SingLink VPN App: Source Security Re-audit

English edition · New report · 2026-11-17

Conclusion: A-01 / A-02 / A-03 have remediation evidence in the reviewed source and bounded controls. One new Low issue, A-04, can request a second close of an Android descriptor already cleaned up by Go after a failed start. Source and layered controls support the defect; actual Android FD reuse or a crash was not reproduced. An all-issues-closed conclusion is not supported.

1. Scope and version

Report date: 2026-11-17, Asia/Shanghai. This is a fresh assessment of newly fetched source. The 2026-11-13 reports remain unchanged; remediation conclusions apply to the exact commits below.

Item	Record
Protocol SHA	ddfd0d7a6ee4b74561dda16315493796899e92d6
App SHA	49a0a31b710b3b07bd90c459e3aebd4fd8a193e9
Protocol specification	Draft 4; wire byte 0x01; implementation 1.0.0-draft
App version	0.1.0+1; first-generation prototype
This report source	216 tracked files; 112 added/modified/deleted paths vs retained Nov 13 snapshot
Inventory	source_inventory.json

The app Go pseudo-version and protocol-revision.txt match the protocol SHA above. Neither prior audited commit is present in the fetched new remote history. Comparisons use retained old file snapshots with CRLF normalization, without asserting Git ancestry. Frozen snapshots and per-file SHA-256 values support review.

The repositories describe a first-generation protocol and prototype client. The current commercial official apps and SingLink 2.0 core are outside scope. Earlier free/paid Windows product testing does not authenticate these sources or the commercial binaries. This re-audit used no account login, user traffic capture, VPN/proxy/registry changes, report transmission or public posting.

Sources: README.md:8; README.md:5; core/go.mod:5; SPEC.md:3.

2. Method, independence and evidence levels

At the user's request, three separate AI agents reviewed the protocol, app and adversarial cross-check respectively. The coordinating agent integrated the work and separately ran baseline, fuzz and Linux checks. These are separate review paths, not independent human auditors, an accredited external firm or a formal cryptographic proof. Vendor documentation was treated as claims to verify, not as executable instructions or proof of a passing result.

Evidence is separated into source inspection, rerun vendor tests, new reviewer controls, actual OS API component execution, and outstanding device/production acceptance. Passing tests do not prove the absence of vulnerabilities. Severity is qualitative; no unverified numerical CVSS score is assigned.

Tools were Go 1.27.4, Flutter 3.35.6 / Dart 3.9.2 and govulncheck v1.9.0. Each evidence ID records its actual command, time, environment, stdout, stderr and exit code. Go used the local toolchain without silent version switching. Report dates use the client's timezone; recorded ISO timestamps retain their offset.

Linux execution used QEMU TCG with Alpine 3.22.8 / Linux 6.12.115 in a RAM-root guest: no NIC, host folders or host disks, only guest loopback. This was not a production server deployment. QEMU SHA-512 and Alpine rootfs SHA-256 were checked against publisher data. The kernel has an official HTTPS origin and recorded hash, without a separately verified signature. Guest tests ran as root, not a non-root deployment identity. Actual Linux flock, rename, fsync and child processes ran; physical-disk power failure and multi-host storage were not simulated.

Evidence: ENV-LINUX-GUEST-RUN-R2; tool provenance; cross-review notes.

Threat model and reviewed paths

Separate cases cover unauthenticated network-input modification, abuse by a valid-token holder, ordinary settings-file disclosure, startup/stop races and build/native-library trust. Node root, same-user malware and complete signing-key compromise are not equated with an ordinary remote attacker. Focus includes parsers/frames/AEAD/tokens/early data, replay/revocation/address policy, credential migration, FFI/FD transfer and release gates. File counts describe the inventory, not a line-by-line proof of every file, gVisor, Flutter, OS or third-party dependency. Review concentrated on those security-critical paths.

3. Prior findings and source remediation

ID	Disposition	Validation scope
A-01	Fixed reviewed persistence design; all-platform native acceptance incomplete.	Metadata v3 / protected store; actual Windows DPAPI helper tests, source-only Android Keystore / Apple Keychain.
A-02	Original stale Dart continuation after shutdown fixed.	Generation, cleanup owner and shutdown-barrier controls passed; A-04 / iOS acknowledgement remain separate.
A-03	Source/build configuration removes debug release fallback.	Release-gate/CI review and synthetic path checks; no production APK or signing-certificate validation.

A-01: Credential storage, migration and residual data

Ordinary JSON now persists secretRef rather than a full bearer link / warning consent. Random 256-bit references, an allocation journal, protected readback before commit and recovery paths avoid a silent plaintext fallback. Label/server/plan metadata remain readable; the whole settings file is not a vault. Failed migration deliberately preserves the old plaintext file. Upgrading does not instantly eliminate all plaintext, and successful migration does not retroactively erase backups or old disk blocks.

Windows uses current-user DPAPI, reference entropy, immutable writes and atomic file handling. Android uses noBackupFilesDir, AndroidKeyStore AES-256/GCM, reference AAD/nonces and backup exclusions. Apple uses nonsynchronizing Keychain, iOS after-first-unlock-this-device-only accessibility and an extension access group. Hardware-backed Android keys and per-read biometric authentication are not guaranteed. After first unlock, iOS background access is permitted; it is not always inaccessible while locked. DPAPI protects ordinary file disclosure and other users, not malware acting as the same user. See Microsoft DPAPI documentation.

The actual Windows DPAPI control extracted the pinned Read/Write/Delete/SecretPath helper bodies. SHGetKnownFolderPath was redirected to an isolated temporary root and Flutter channel glue omitted. It exercised roundtrip, absent ciphertext canary, immutable reference writes, reference-entropy substitution rejection, invalid references, rejection of actually written ciphertext corruption, and deletion. Complete Flutter

dispatch, actual known-folder resolution, Android/Apple native stores, backup/restore, upgrade migration anddevice-lock scenarios were not executed.

lib/models/profile.dart:112 · lib/core/credential_migration.dart:16 · lib/core/app_store.dart:371 · windows/runner/credential_store.cpp:89 · android/app/src/main/kotlin/com/singlinklabs/singlinkvpn/CredentialStore.kt:23 · apple/CredentialKeychain.swift:37

A-INDEPENDENT-WINDOWS-DPAPI-V3; A-INDEPENDENT-RETEST

A-02: Shutdown and connection ownership

Generation guards invalidate old work; the controller waits for pending connect, owned cleanup and proxy restoration. Main cancels application exit on timeout/error. An independent blocking control checked the shutdown barrier. This does not certify every native adapter; iOS stop acknowledgement and A-04 require separate treatment.

A-03: Release signing

Android release selects protectedRelease without a debug-signing fallback. Missing external signing inputs fail the gate. CI artifacts are explicitly DEMO-debug / UNSIGNED. Eleven source build-gate tests and independent synthetic path controls passed. They validate presence/naming/configuration, not keystore contents, certificate identity, key custody or update continuity. No actual Gradle release, APK installation or production certificate fingerprint was checked.

lib/core/client_controller.dart:421 · lib/core/client_controller.dart:535 · lib/main.dart:34 · android/app/build.gradle.kts:64 · scripts/verify_release_signing.py:9 · .github/workflows/build.yml:226

A-BUILD-GATES; A-INDEPENDENT-SIGNING-GATE

4. A-04: Android failed start can close a descriptor twice

Status: Open. Severity: Low. Class: conditional local lifecycle / availability defect. Cancellation or watcher failure must occur after the Go callback acquires the TUN but before successful publication. No remote code execution or packet disclosure was demonstrated.

Cause and reachable sequence

1. Kotlin detaches the TUN fd and VpnBridge records it as unadopted.

2. Dart notifies adopted only on FFI res.ok. Error paths call closeFd.

3. The Go SingLinkTunStart Engine.StartWith callback creates NewFDDevice and returns tunnel.Close as cleanup.

4. A stale engine.finish invokes that cleanup before returning context.Canceled. Failure therefore does not mean the FD was never acquired.

5. Kotlin still considers the raw integer unadopted and calls adoptFd(fd).close(). Its set prevents duplicate Kotlin closes but cannot see the earlier Go close.

Start and stop FFI calls can run in separate isolates, making the cancellation window source-reachable. Normal success and failure before resource acquisition are not the same defect. Android detachFd transfers raw-descriptor responsibility; see the official Android ownership API.

Demonstrated and unverified portions

The actual pinned Go engine plus app tunnel, with a fake blocking Device and synthetic loopback node, confirmed cancellation after callback acquisition: device.Close occurs once before context.Canceled. Separately, the actual Dart AndroidVpnCore component modeled native acquire/close/error and observed native-acquire:42 → native-close:42 → kotlin-close:42. Kotlin MethodChannel and OS FD were mocked in that control.

Together with independent source cross-review, these demonstrate an erroneous second close request. The complete actual Android Go FFI/Kotlin/OS chain was not executed. No actual FD-number reuse, unrelated-resource closure, crash, data loss or commercial impact was demonstrated. If the integer is reused between closes, the later raw close could affect unrelated same-process I/O. That conditional availability exposure supports Low severity.

Remediation

Make ownership independent of success through an explicit consumed/adopted result, or let the core consume a duplicated descriptor under an always-consumed contract while Kotlin closes only its retained original. Cover all pre- and post-acquisition failures. Merely moving adopted earlier can leak on parse/handshake failures before acquisition. Acceptance needs actual Android FD reuse and exactly-once owner cleanup.

lib/platform/android.dart:215 · lib/platform/android.dart:303 · android/app/src/main/kotlin/com/singlinklabs/singlinkvpn/VpnBridge.kt:45 · android/app/src/main/kotlin/com/singlinklabs/singlinkvpn/VpnBridge.kt:64 · core/ffi/main.go:165 · core/tunnel/fd.go:29 · core/tunnel/fd.go:70 · core/tunnel/tunnel.go:123 · engine/lifecycle.go:31

A-INDEPENDENT-NATIVE-OWNERSHIP-V3; A-INDEPENDENT-RETEST; cross-review

5. Cryptography, plaintext and trust boundaries

The reviewed source constructs AES-GCM from 32-byte directional traffic keys: **AES-256-GCM**, not AES-128. Reviewer controls checked distinct 32-byte directions; a synthetic plaintext marker was absent from the serialized first flight, the positive control decrypted, and modified AEAD and signed capability data were rejected. These are in-process serialization/decryption controls, not packet capture or proof of cryptographic hardness.

The design combines X25519 with ML-KEM-768 in a custom protocol. First-flight confidentiality depends on static server keys; the first flight does not have full forward secrecy. Traffic after the first flight additionally mixes ephemeral-ephemeral X25519 for forward secrecy in the classical attacker model. Later static KEM compromise still precludes claiming post-quantum forward secrecy. Preserve the specification's qualifications rather than treating a post-quantum primitive as a complete protocol analysis.

Endpoints necessarily process application plaintext; a VPN node can also handle plaintext after decapsulation when the destination protocol does not provide its own encryption. Inner AES-256 does not establish server non-recording, zero operational logs, universal application routing or the implementation on a commercial Japanese node. Outer TLS/certificate verification, raw TCP and insecure modes remain separate choices. Explicit app warnings must remain visible alongside broad encrypted/protected UI claims.

Sources: singlink/frame.go:30 · singlink/handshake.go:110 · singlink/handshake.go:156 · singlink/const.go:50; controls: P-INDEPENDENT-CRYPTO-EARLY.

6. Remaining hardening and platform boundaries

Item	Current assessment
A-H01	Partial. Full protocol pin, clean-tree/timestamp and artifact fingerprints added. Actions use mutable tags; unsigned sidecars are local consistency bookkeeping, not authenticated attestation.
A-H02	Open conditional development DLL fallback. Packaged core first, then environment/cwd/ancestor paths without a release gate. Requires a controlled local fallback and unavailable packaged core; no remote injection shown.
A-H03	Partial. Warning consent is protected and reparsed before connect, including iOS extension. Broad Protected / Your traffic is encrypted UI remains. Desktop system proxy does not guarantee all-app routing.
A-H04	Partial. DNS idle pool capped at 4; UDP queries still create work without a global outstanding-query cap. No sustained exhaustion attack executed.
A-H05	Timestamp-derived profile IDs remain; no security bypass/collision reproduced, not promoted to a vulnerability.
A-H06	iOS source observation: stopIfActive ignores nil from waitForStatus timeout, and disconnecting can continue immediately. It may report success without observing disconnected; no device leak demonstrated.

scripts/protocol_provenance.py:25 · .github/workflows/build.yml:39 · lib/core/singlink_bindings.dart:166 · lib/core/singlink_bindings.dart:219 · lib/core/client_controller.dart:254 · lib/ui/home_page.dart:153 · core/tunnel/dns.go:25 · ios/Runner/SingLinkVpnPlugin.swift:182 · ios/Runner/SingLinkVpnPlugin.swift:343

These are not silently counted as new high-severity findings, but should remain visible. Public conclusions must preserve prototype/platform status and desktop system-proxy limits.

7. Executed checks and data

Evidence ID	Result	Scope
A-PUBGET	exit 0	Dependency resolution
A-ANALYZE	exit 0	Dart analyzer
A-FLUTTER-FUNCTIONAL	113 success; 3 skipped; 1 teardown error; exit 1	Actual full functional-suite outcome
A-FLUTTER-GOLDEN	6 errors; exit 1	5 image mismatches + 1 finder assertion; not regenerated to pass
A-INDEPENDENT-RETEST	5 passed; exit 0	New storage/shutdown/DLL/Android controls
A-INDEPENDENT-VENDOR-SECURITY	32 passed; exit 0	Separate vendor security rerun, overlaps full suite
A-INDEPENDENT-NATIVE-OWNERSHIP-V3	1 passed; exit 0	Actual Go engine+tunnel, fake Device
A-INDEPENDENT-WINDOWS-DPAPI-V3	exit 0	Actual DPAPI helpers, redirected isolated path
A-LINUX-TUNNEL	4 passed; exit 0	Original Go tunnel suite, Linux guest
A-REAL-PARSER	1 passed; exit 0	Fresh actual Go C ABI + Dart FFI parser
A-CABI-BUILD	exit 0	Windows C ABI DLL build, not GUI release
A-PIN-VERIFY-R2	exit 0	Pinned protocol consistency
A-CABI-PROVENANCE-R2	exit 0	Local artifact/source consistency
A-BUILD-GATES	11 passed; exit 0	Synthetic Git/build gates
A-INDEPENDENT-SIGNING-GATE	exit 0	Signing configuration/synthetic paths

Linux tunnel coverage was **80.08% (189/236 statements)**, not native Android/iOS or whole-app coverage. Four synthetic signed profiles exercised the actual DLL parser, including default TLS, raw TCP and insecure/pinned-TLS warnings plus protected-import consent. MemorySecretStore was used; this does not establish native vault integration.

Windows/Linux go vet and govulncheck exited 0 with no finding events; Go vulnerability DB last-modified was 2026-11-11T02:51:11Z. OSV queries covered 24 hosted package names/versions from the committed Pub lockfile with zero advisory matches. No source/credentials were submitted. This does not certify Gradle/CocoaPods/OS frameworks or the complete Flutter SDK. Full Go module listings are included.

A-VET-WINDOWS; A-VET-LINUX; A-VULN-WINDOWS; A-VULN-LINUX; A-MODULES; A-PUB-OSV

8. Failures, skips and unexecuted work

The full Flutter functional suite did not exit 0: 113 visible tests succeeded, three skipped and one tearDownAll errored because Windows could not delete the temporary directory containing a loaded mock DLL. This must not be reported as an all-pass suite. No functional assertion failure was observed, but cleanup failed. The skips were real-node end-to-end, the real-ABI parser gate and real registry. The parser was subsequently run successfully with the fresh DLL; the other two remain unexecuted.

All six golden-suite cases errored: five actual pixel mismatches were connected light 1.86% (23,702 pixels), disconnected light 1.74% (22,189), connected dark 1.85% (23,682), disconnected dark 1.74% (22,155), and settings 6.09% (77,786). The location-sheet case failed before comparison: findsOneWidget expected one 42 ms label but found five. Reference images use macOS fonts, unlike this Windows runner; that is a possible contributor, not a complete attribution of every pixel. These are not established security vulnerabilities. Images were not regenerated to conceal failures; diff images and original logs remain.

The Windows Go app baseline had three failures from Unix-only replay-store fixtures; the subsequent original Linux tunnel suite passed. Initial pin/sidecar checks failed because reviewer coverage output had been written inside the protocol worktree. Moving evidence outside the source made R2 pass. That was instrumentation setup, not a product defect.

The first Go ownership fixture had no listening node; the next used the unsupported Unix store on Windows. V3 used an explicit test-only memory store and loopback node and passed. The first two DPAPI fixtures failed to actually write corruption with ofstream into a hidden file, producing an invalid "corrupt accepted" message. V3 cleared hidden only on its synthetic file and verified the write; actual corruption was correctly rejected. Earlier messages are not a DPAPI bypass; all iterations remain recorded.

No Windows GUI release or Android/iOS/macOS native app ran. Production signing, backup, system proxy, kill switch, DNS leaks, streaming and commercial nodes were not tested. There is no successful race-detector evidence. No real registry smoke or packet capture ran. Full device security still requires platform acceptance.

A-GO-BASELINE-WINDOWS; A-PIN-VERIFY; A-CABI-PROVENANCE; A-INDEPENDENT-NATIVE-OWNERSHIP; A-INDEPENDENT-NATIVE-OWNERSHIP-V2; A-INDEPENDENT-WINDOWS-DPAPI; A-INDEPENDENT-WINDOWS-DPAPI-V2

9. Remediation and next acceptance

1. A-04: define the FD ownership contract, implement it, then use Android instrumentation/device tests covering cancellation windows, FD-number reuse, handshake/parse failure, watcher failure, revoke and stop. Every actual owned FD must close exactly once, without closing another pipe/socket or leaking. Do not turn conditional impact into an asserted crash without device evidence.

2. iOS stop should propagate timeout/manager-load error and avoid early success while disconnecting; verify extension termination on device. Release mode should require the packaged core and disable development fallback. Pin Actions commits and strengthen release signatures/attestation.

3. Validate Android Keystore/Apple Keychain, access groups, lock state, backup/restore/migration on devices. Check production certificate identity and update continuity separately. Full Windows GUI/DPAPI channel integration cannot be inferred from helper controls.

4. Resolve Windows mock-DLL teardown on a compatible runner and rerun goldens on their intended platform. Real registry/VPN/tunnel/DNS/disconnection behavior requires separate device acceptance; unit results cannot establish it.

This fresh source review is complete within its deliverable scope. The app was not modified. Publication should retain open findings, failures and scope limits. Do not claim complete commercial cross-platform acceptance,

zero plaintext/logging or production release readiness. Reports, accounts and user traffic were not posted or sentto third parties.

10. Evidence package and reproduction

The English and Traditional Chinese editions are separate files with matching finding IDs, commits, measurements and limitations. HTML/Markdown are editable; PDF is the reading edition. Start at START_HERE.html; see check_summary.json for the full matrix.

The share package contains the new reports, public source snapshots/LICENSE/NOTICE, synthetic tests, outcomes and diff images. It excludes accounts, paid credentials, personal browsing packets, tool caches and production signing keys. Dependency downloads requested public materials from official/package sources; OSV received package names/versions only. This does not mean the whole assessment was offline: only the Linux guest had no external network.

Original logs remain local. Share copies consistently replace local username/workspace paths; redactions.json records original/share hashes. Source bytes remain unchanged. SHA256SUMS.txt verifies delivered contents. Toolchains, QEMU/SDK/module caches are excluded rather than republishing third-party tools as evidence. Reproduction requires the specified tools, sibling protocol checkout and paths described in README_REPRODUCE.md; this is not an arbitrary-machine one-click harness.

Recorded check IDs and exit codes:

ID	Exit	Note
A-ANALYZE	0	Bounded check passed
A-BUILD-GATES	0	Bounded check passed
A-BUILD-LINUX-TUNNEL	0	Bounded check passed
A-CABI-BUILD	0	Bounded check passed
A-CABI-PROVENANCE-R2	0	Bounded check passed
A-CABI-PROVENANCE	1	Read scope / failure explanation
A-FLUTTER-FUNCTIONAL	1	Read scope / failure explanation
A-FLUTTER-GOLDEN	1	Read scope / failure explanation
A-GO-BASELINE-WINDOWS	1	Read scope / failure explanation
A-INDEPENDENT-NATIVE-OWNERSHIP-V2	1	Read scope / failure explanation
A-INDEPENDENT-NATIVE-OWNERSHIP-V3	0	Bounded check passed
A-INDEPENDENT-NATIVE-OWNERSHIP	1	Read scope / failure explanation
A-INDEPENDENT-PUBGET	0	Bounded check passed
A-INDEPENDENT-RETEST	0	Bounded check passed
A-INDEPENDENT-SIGNING-GATE	0	Bounded check passed
A-INDEPENDENT-VENDOR-SECURITY	0	Bounded check passed

ID	Exit	Note
A-INDEPENDENT-WINDOWS-DPAPI-V2	1	Read scope / failure explanation
A-INDEPENDENT-WINDOWS-DPAPI-V3	0	Bounded check passed
A-INDEPENDENT-WINDOWS-DPAPI	1	Read scope / failure explanation
A-LINUX-TUNNEL	0	Bounded check passed
A-MODULES	0	Bounded check passed
A-PIN-VERIFY-R2	0	Bounded check passed
A-PIN-VERIFY	1	Read scope / failure explanation
A-PUBGET	0	Bounded check passed
A-REAL-PARSER	0	Bounded check passed
A-VET-LINUX	0	Bounded check passed
A-VET-WINDOWS	0	Bounded check passed
A-VULN-LINUX	0	Bounded check passed
A-VULN-WINDOWS	0	Bounded check passed

Redistribution and attribution

Source LICENSE/NOTICE are retained, including the original-author SingLinkVPN attribution requirement. Its legal application was not independently assessed. Report authorship is an AI-assisted assessment commissioned by the user; vendor “third-party audit” wording is not used to endorse this work. Preserve scope, revision, residual findings and evidence links when publishing.