

HostSpica security design report, version 1

HostSpica, version 1, 2026-10-04. Applies to HostSpica Identity 1.0.0. hostspica.com/research/hostspica-security-design-report-v1

What does the HostSpica security design report say? It describes how HostSpica Authenticator, Passkey and Identity keep secrets on the phone with Android Keystore keys, biometric checks and no internet permission; what we tested; and what we know is weak, including no independent audit, no FIDO certification and a clipboard clear that is less reliable than described. It is our own report for HostSpica Identity 1.0.0.

1. Scope and status

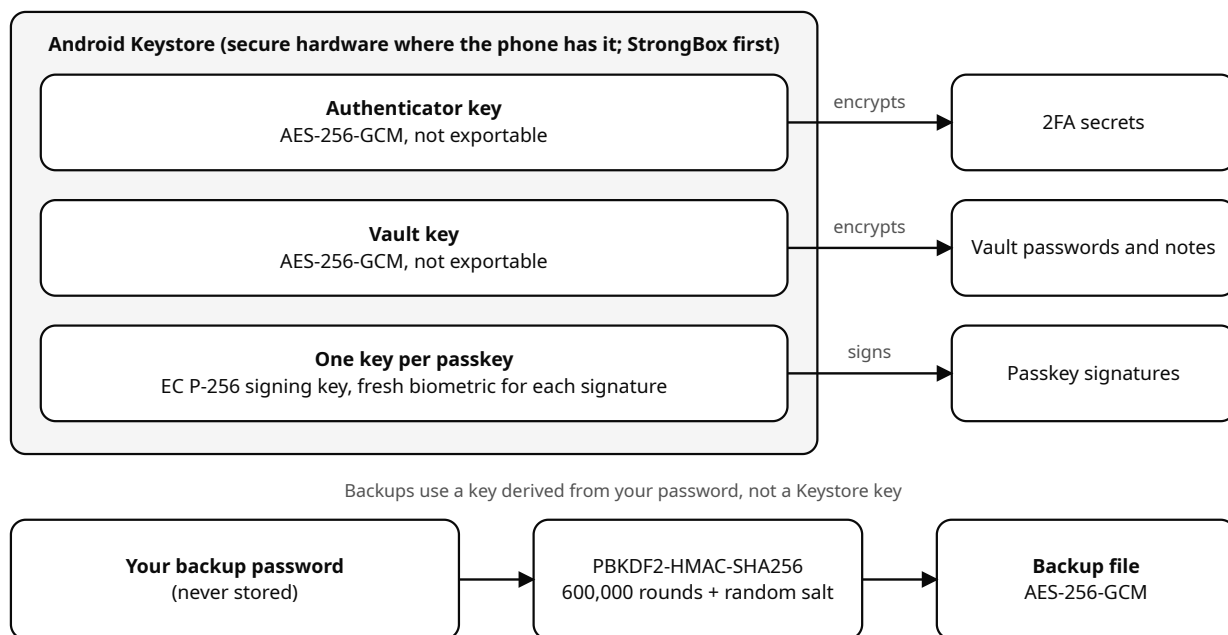
This report covers HostSpica Identity 1.0.0 for Android, which contains the Authenticator, Passkey, AirAuth and Vault features, and the standalone HostSpica Authenticator and Passkey apps where they share code. It was written on 4 October 2026 by the people who built the apps. **It is our own assessment. It is not an audit, and no independent party has reviewed it.** HostSpica Passkey is not FIDO certified.

2. What we protect

- 2FA secrets and Vault passwords and notes (confidentiality).
- Passkey private keys (confidentiality and non-exportability).
- The act of signing in and of filling a password (only you can approve it).
- Backup files (confidentiality and tamper detection).

3. Architecture in brief

Each feature stores its data in its own local database inside the app's private storage. Secrets are encrypted with AES-256-GCM under keys that Android Keystore holds; passkeys use one hardware-backed P-256 signing key each. The app declares no internet permission, so it cannot send data over the network itself. Details: [where every secret lives](#).



The keys in HostSpica Identity and what each one protects.

4. Controls and their evidence

Controls as of 1.0.0.

Control	How it works	Evidence	Limit
No network access	No INTERNET permission in the merged manifest	Checked with aapt2 on the release build; method published	Android's own cross-device passkey flow is run by the system, not by the app
Secrets at rest	AES-256-GCM, Keystore keys, separate keys for Authenticator and Vault	Code review; encrypt and decrypt tests	Labels (issuer, title, username, site) are stored unencrypted for search
App lock	Fingerprint, face or screen lock on open; configurable time in background	Manual device checks of both Immediately and timed modes	A known screen-lock PIN also unlocks
Passkey signing	Per-key Keystore signing, fresh biometric for each signature, origin from privileged browsers	Registered and signed in on Google, GitHub, Microsoft and webauthn.io	ES256 only; native-app callers not reviewed
Autofill	Placeholders only; real values released after a biometric check; exact registered domain on web	Unit tests for lookalike domains; manual save and fill checks	No gate if the phone has no screen lock
Backups	PBKDF2-HMAC-SHA256 600,000 rounds, AES-256-GCM	Round trip, wrong password and tamper tests	Password strength is the whole defence
Screen capture	Secure-window flag on the app window	Manual check that screenshots are blocked	Camera or accessibility capture not stopped

Clipboard	Sensitive flag and 30-second clear for Vault copies	Manual checks	Authenticator code copies are not reliably cleared; see below
-----------	---	---------------	---

5. Testing

- **Automated:** 80 JVM unit tests (as of 3 October 2026): the RFC 4226 and RFC 6238 test vectors, Google export decoding, backup crypto, site matching, authenticator data layout and signing, passkey request handling.
- **Manual, on one Android 16 phone:** Autofill save and fill, backup and restore with wrong and right passwords, the release build with R8, passkey registration and sign-in on Google, GitHub, Microsoft and webauthn.io with Chrome and Brave, refusing duplicate passkeys on webauthn.io and GitHub, the standard cross-device flow listing HostSpica Identity.
- **Not tested:** other phones and Android versions; Firefox and Edge; a complete sign-in on a computer through Android's cross-device flow; penetration testing; fuzzing; side-channel analysis.

6. Known weaknesses

1. No independent security audit.
2. Not FIDO certified; our own AirAuth Bluetooth protocol is not a FIDO transport and is unreviewed.
3. The 30-second clipboard clear for Authenticator codes depends on the screen staying open (see [clipboard and screenshot protections](#)).
4. Unencrypted labels in the private database.
5. Passkey keys may be invalidated by Android if a new fingerprint is enrolled; untested.
6. A compromised or rooted phone defeats on-device protections.
7. ES256 is the only passkey algorithm, and native-app callers have not been reviewed in detail.

7. Changes since design

Security-relevant changes are listed in the [security changelog](#), including the Autofill domain-match fix, the passkey request-handling changes and the removal of our custom Bluetooth passkey flow.

8. Reporting a problem

See the [disclosure policy](#). Write to contact@hostspica.com.

Download this report as a PDF: [hostspica-security-design-report-v1.pdf](#).

References

- OWASP MASVS: <https://mas.owasp.org/MASVS/>
- Android Developers: Android Keystore system: <https://developer.android.com/privacy-and-security/keystore>
- W3C Web Authentication Level 2: <https://www.w3.org/TR/webauthn-2/>
- NIST SP 800-63B: <https://pages.nist.gov/800-63-3/sp800-63b.html>