ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IX, September2026

Passkeys in Multi-Device Ecosystems: Reference Architecture, Recovery, Telemetry, and Phishing-Resistant Authentication at Scale

Sasibhushana Matcha[1], Sreekanth Rachamadugu[2]

[1]Visvesvaraya Technological University, Machhe, Belagavi, Karnataka 590018, India

[2]MetLife

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150900114

Received: 13 February 2026; Accepted: 18 February 2026; Published: 19 October 2026

ABSTRACT

Abstract—Passkeys extend WebAuthn/FIDO2 beyond a single browser API into a multi-device capability. The cryptography is mature, but deployments often struggle at the edges: device changes, recovery, cross-device flows, and inconsistent behavior across web and native surfaces. This paper provides a practical reference architecture, end-to-end flows, and an operational playbook (telemetry, SLOs, and failure taxonomy). We include a scenario-based evaluation using synthetic traces and de-identified event schemas so teams can adapt the approach to their environment.

Keywords: Passkeys, WebAuthn, FIDO2, platform authenticators, multi-device sync, cross-device authentication, account recovery, risk-based policy, telemetry, CIAM, phishing resistance, incident response.

INTRODUCTION

Passkeys replace shared secrets with device-held private keys and public-key verification. That change raises the security floor, but it also shifts the operational burden: teams must manage enrollment, sync, recovery, and user experience across devices and apps.

Most practical failures come from lifecycle boundaries (new phone, lost device, account recovery), gaps between web and native implementations, and the long tail of platform or browser differences. Without good instrumentation, debugging becomes anecdotal and products drift back to weaker fallbacks.

Contributions: (1) a reference architecture that treats multi-device trust, recovery, and policy as first-class concerns; (2) end-to-end flows including cross-device approval and step-up; (3) recovery patterns that preserve phishing resistance; (4) a telemetry and incident-response playbook grounded in a failure taxonomy; and (5) a small reproducibility kit for scenario analysis.

Background and Related Work

A. From WebAuthn to Passkeys. WebAuthn and FIDO2 define how browsers/apps create credentials and produce signed assertions. Passkeys extend this by making credentials discoverable and portable across a user's device set via platform sync, enabling username-less login and cross-device flows.

B. Phishing-Resistant MFA. Prior work highlights that OTP and push approvals remain phishable via relay attacks and social engineering. Passkeys reduce these risks by binding assertions to RP IDs and verified origins, but require careful fallback design to avoid reintroducing weak factors.

C. Deployment Experience. Industry deployments show that usability, recovery, and device enrollment drive real adoption. As a result, a practical passkey system must define policy, telemetry, and recovery as first-class components—not optional add-ons.

D. Gap Addressed. Existing descriptions often stop at protocol mechanics. This work focuses on operating passkeys at scale: lifecycle, interoperability, risk-based policy, and measurable reliability.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IX, September2026

Passkey Ecosystem, Sync, and Trust Model

Fig. 1 frames passkeys as an ecosystem capability spanning user devices, platform sync providers, and the relying party (RP). The RP verifies signatures and enforces policy, while platforms protect private keys and mediate user verification (biometric/PIN).

Components

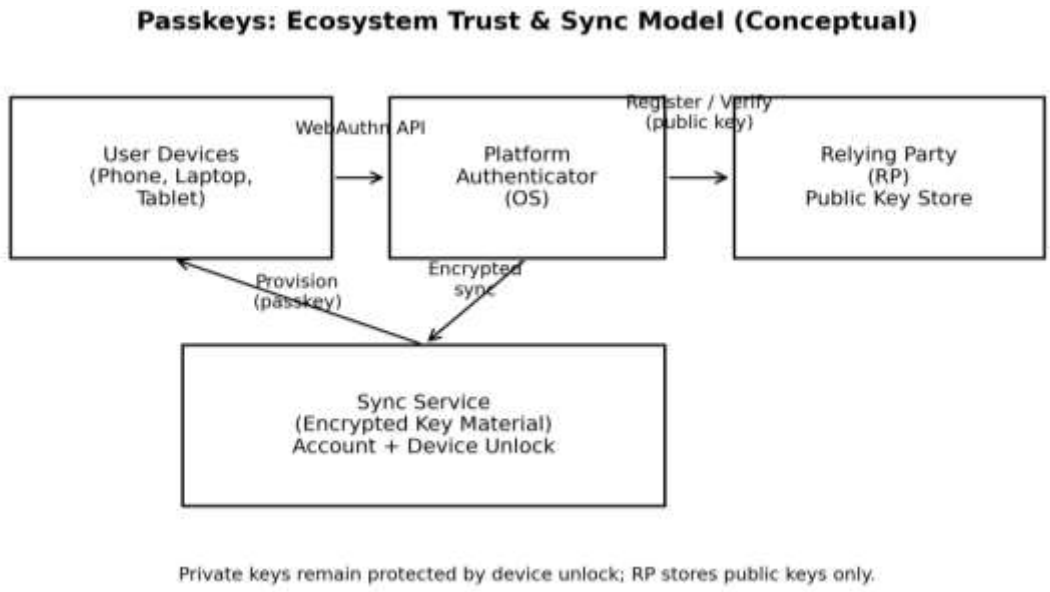
1) User devices: phones, laptops, tablets; browsers and native apps invoking WebAuthn.
2) Platform authenticators: secure hardware/TEE-backed key storage and UV/UP enforcement.
3) Sync providers: platform keychains that replicate encrypted credential material across a user's devices.
4) RP services: credential registry, challenge service, and policy/telemetry pipeline.

Trust Boundaries. The RP trusts the authenticator to enforce user verification and origin binding; the authenticator trusts the platform to secure keys and to gate sync. The sync provider is trusted for availability but should not be trusted with plaintext private keys; end-to-end encryption and hardware-backed keys reduce that exposure.

Identity Binding. Credential IDs and public keys are bound to an account at the RP. Account linking must be resilient to username changes, email aliasing, and multi-tenant identities by using stable internal subject identifiers.

Attestation and Device Posture (Optional). Attestation may support high-assurance use cases (regulated access, admin actions) but increases privacy risk and operational complexity. A pragmatic approach is "attestation-on-demand" combined with device posture signals (OS version, screen lock, jailbreak/root).

Fig. 1. Multi-device passkey ecosystem and trust boundaries (original diagram).



End-To-End Flows: Registration, Assertion, and Cross-Device Login

A. Registration (Create). The RP issues a challenge and policy (RP ID, allowed algorithms, resident-key requirement). The client invokes WebAuthn create; the authenticator performs user verification and returns attested or non-attested credential material. The RP validates origin, challenge, and public key; then stores credential metadata (credential ID, public key, sign counter, AAGUID if used).

B. Assertion (Get). The RP issues a challenge and allowed credential list (or requests discoverable credentials for username-less login). The authenticator signs the challenge with the private key under RP ID constraints. The RP verifies the signature, checks sign-counter semantics, evaluates risk policy, and issues a session.

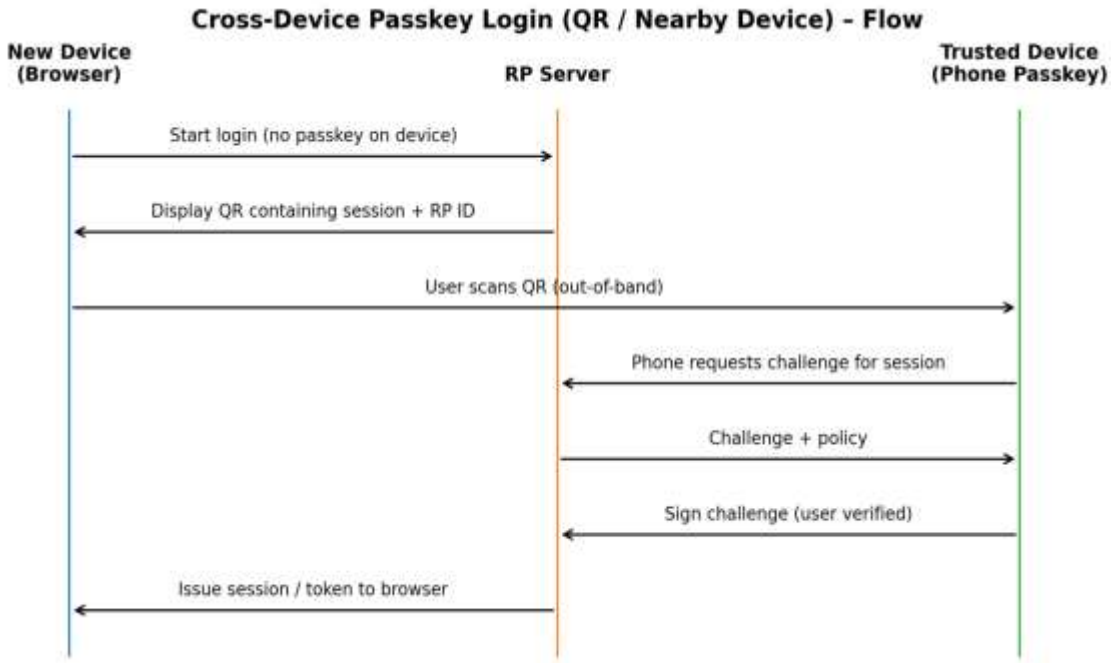
ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IX, September2026

C. Step-Up and Transaction Confirmation. For sensitive actions (payment, profile change, device enrollment), the RP can require UV=required and include a transaction binding text (or structured data) so the user confirms intent at the authenticator surface.

D. Cross-Device Login. When a user logs in on a new device without a local passkey, a nearby trusted device can approve and sign via a short-range assisted flow (e.g., QR/nearby). The RP must treat the resulting assertion as high assurance only if the assisting device meets posture requirements and UV is performed.

E. Discoverable Credentials. Passkeys are typically resident/discoverable, enabling username-less login. The RP should still provide account selection UX to prevent confusion when multiple accounts exist for the same user.

Fig. 2. Cross-device passkey assertion via nearby-device approval (original diagram).



Credential Lifecycle: Management, Recovery, and Delegation

A. Lifecycle States. Credentials move through: created → verified-in-use → rotated/augmented → revoked. Maintain per-credential metadata (first/last use, device class, UV capability, last known posture) to support risk policy.

B. Device Change and "New Device" Enrollment. Prefer enrollment driven by an existing passkey assertion (step-up) rather than SMS/email. If weaker recovery must exist, rate-limit and require additional proofing before allowing passkey creation.

C. Recovery Patterns. Provide at least two recovery routes: (1) device-to-device transfer/nearby approval; (2) recovery codes or hardware key. If using help-desk recovery, log a cryptographic audit trail and enforce cooling-off periods before high-risk actions.

D. Delegated Access and Shared Devices. For enterprise or family scenarios, avoid sharing passkeys. Use session delegation (limited-scope tokens) or managed device accounts; for kiosks, require ephemeral sessions and explicit sign-out with device attestation where applicable.

E. Key Material and Algorithms. Prefer modern algorithms (e.g., ES256) with server-side algorithm agility. Store public keys and metadata in a tamper-evident store; encrypt at rest and separate auth data from user profile data.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IX, September2026

SECURITY ANALYSIS: THREATS, CONTROLS, AND RESIDUAL RISK

A. Threat Model. Attackers may phish users, compromise passwords from other sites, replay OTPs via relay, intercept network traffic, or breach RP databases. We also consider device malware and account recovery abuse. The goal is to prevent scalable credential replay and reduce single-point recovery failures.

B. Phishing and Relay Resistance. WebAuthn assertions are scoped to RP IDs and validated origins, limiting classic phishing. Residual risks include social engineering into approving cross-device logins; mitigate with clear transaction text, device posture checks, and anomaly detection on assisted flows.

C. Database Breach. Stolen public keys and credential IDs are insufficient to authenticate without private keys. Protect against session theft with secure cookies, token binding, short TTLs, and refresh-token rotation.

D. Device Compromise. If the authenticator is compromised, local UV/UP may be bypassed. Mitigate via device posture signals, attestation-on-demand for high-risk actions, and rapid revocation. Provide user-visible device inventory and one-click credential revocation.

E. Recovery Abuse. Recovery is the primary downgrade channel. Apply rate limits, step-up requirements, cooling-off, and audit. Treat recovery as a security product, not a support afterthought.

Privacy, Accessibility, and Human Factors

A. Privacy by Design. Minimize stored metadata, avoid stable device fingerprints, and treat AAGUID/attestation data as sensitive. Use purpose limitation: collect telemetry needed for reliability and fraud detection, but aggregate and retain minimally.

B. Accessibility. Passkeys can improve accessibility by removing typing burdens. Provide clear fallbacks, localized prompts, and support for assistive technologies. Avoid dark patterns that push users into weaker MFA.

C. UX Principles. Communicate “what is happening” during cross-device approvals; display relying-party identity and action context; and provide transparent device management so users can understand and revoke credentials.

Telemetry, Reliability, and Incident Response

A. Instrumentation Model. Log structured events for create/get ceremonies, including stage timings (challenge, client invocation, authenticator, server verify), failure codes, and policy decisions. Separate security telemetry from product analytics to enforce least-privilege access.

B. SLOs and Key Metrics. Track completion rate by surface (web/native), median and tail latency, assisted-flow rate, recovery volume, and credential revocation frequency. Monitor step-up success and abandonment to detect friction regressions.

C. Failure Taxonomy. Normalize failures into buckets: RP policy (origin/RP ID mismatch), authenticator availability, user cancellation, platform UI errors, network/clock skew, and risk engine denials. This enables targeted fixes rather than broad rollbacks.

D. Progressive Delivery. Use feature flags by cohort, platform, and risk segment. Maintain “safe fallback” paths and kill-switches for assisted flows separately from local passkey flows.

E. Incident Playbooks. For spikes in assisted approvals or unusual revocation patterns, trigger automated containment: tighten policy, require UV, or temporarily disable cross-device approvals. For suspected compromise, rotate tokens, force re-auth with passkeys, and notify affected users.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IX, September2026

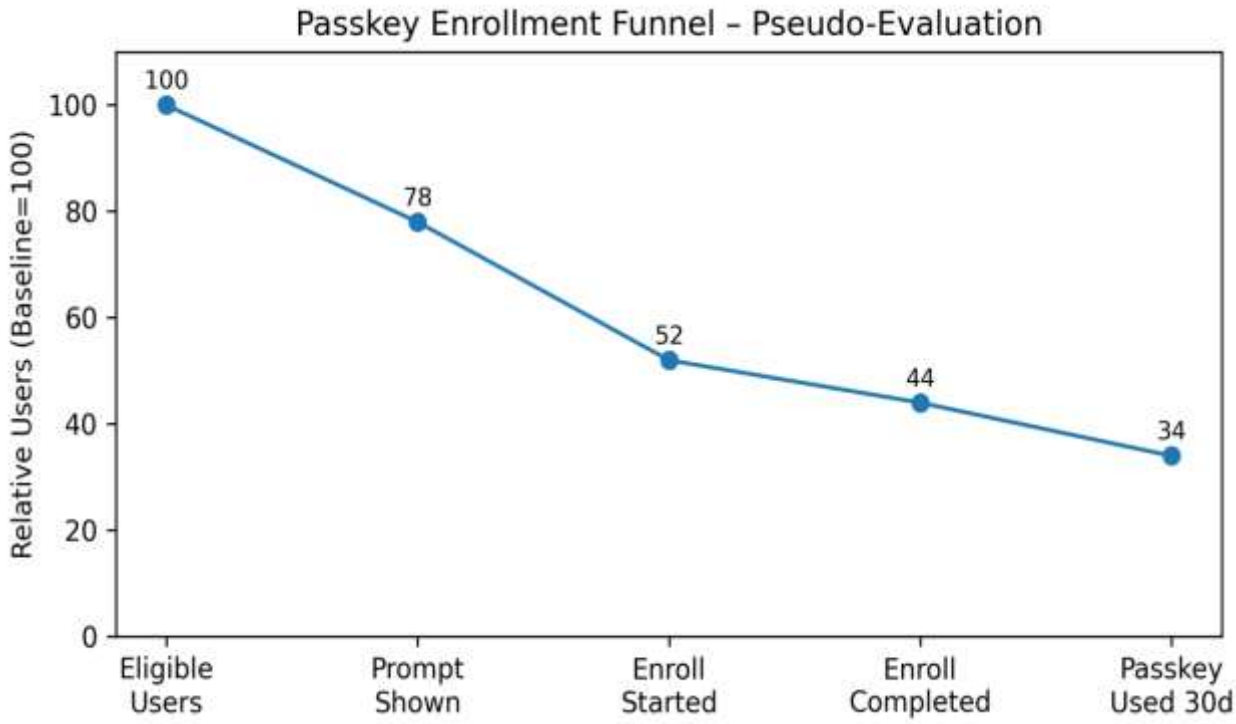


Fig. 3. Passkey enrollment and usage funnel with progressive rollout cohorts (pseudo-evaluation, original graph).

Deployment Playbook: Migration and Interoperability

A. Readiness Checklist. Ensure stable account identifiers, session hardening, device inventory UI, and recovery controls before enabling passkeys broadly.

B. Migration Strategy. Start with optional enrollment for low-risk users, then expand with nudges after successful logins. Avoid forced enrollment until telemetry shows stability.

C. Interop Considerations. Test across major browsers and OS versions; ensure RP ID alignment across subdomains; handle multiple authenticators per user; and validate behavior with password managers and platform sync providers.

D. Policy Governance. Define clear rules for when passkeys are required vs optional, what step-up means, and how recovery is audited. Treat policy as code with reviews and change logs.

Evaluation Design and Reproducibility

A. Evaluation Goal. We evaluate reliability and security outcomes of a passkey program under realistic heterogeneity: mixed devices, mixed browsers, and varying network conditions.

B. Replication Package. Because real user authentication telemetry may be sensitive, we define a de-identified event schema and a synthetic trace generator that preserves timing and failure distribution characteristics while removing PII.

C. Workloads. Synthetic cohorts simulate consumer and workforce IAM scenarios: primary login, step-up, cross-device login, and recovery. Each cohort varies device posture, sync availability, and user cancellation behavior.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IX, September2026

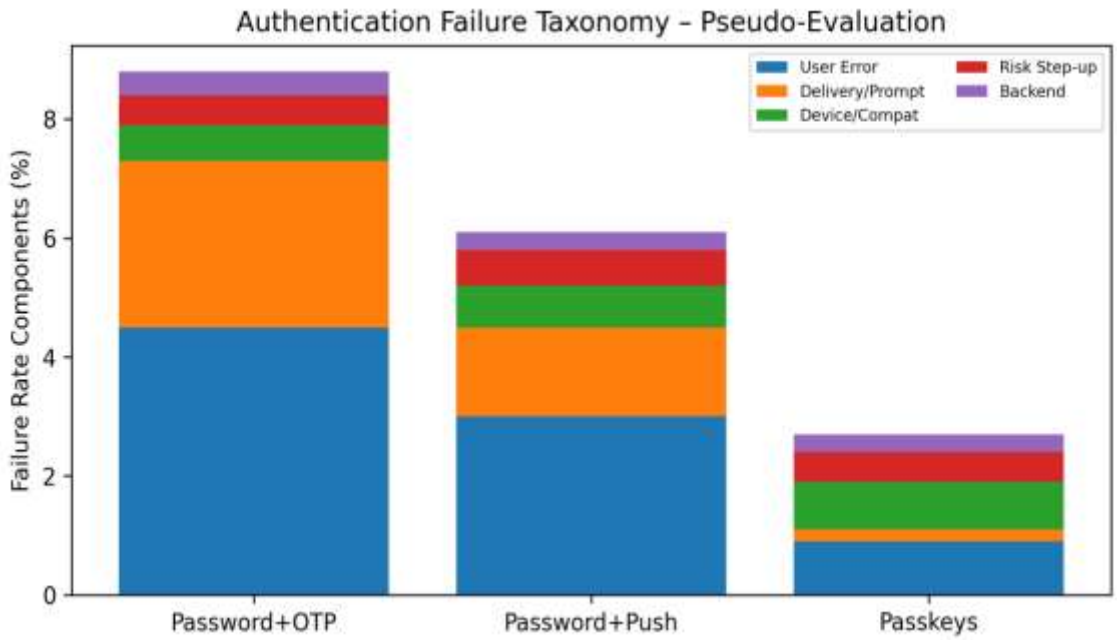


Fig. 4. Authentication failure taxonomy normalized across methods (pseudo-evaluation, original graph).

D. Metrics. Completion rate, assisted-flow ratio, median/P95 latency, recovery volume, and simulated attack success rate (phishing relay, credential stuffing, recovery abuse) are reported. Baselines compare password+OTP and password+push approval with equivalent session hardening and rate limits.

Illustrative Evaluation: Scenario-Based Results and Discussion

A. Scale Assumptions. The pseudo-evaluation models 8–15 million active users across web and native apps, with 60–110 million login attempts per month, and a staged rollout from 5% to 80% of eligible users over 10 weeks.

B. Completion Rate. Passkeys increase successful login completion from ~95% (password+OTP) to ~98.7% after rollout stabilization, primarily by reducing OTP delivery failures and password-related lockouts. Assisted cross-device flows show slightly higher abandonment when prompts are unclear, motivating improved UX copy and posture-aware policy.

C. Enrollment Funnel. Offering enrollment immediately after a successful login (with clear value framing) yields materially higher opt-in than offering on failure paths. Multi-device coverage (≥2 devices with passkeys) is a leading indicator of long-term durability because it reduces recovery events.

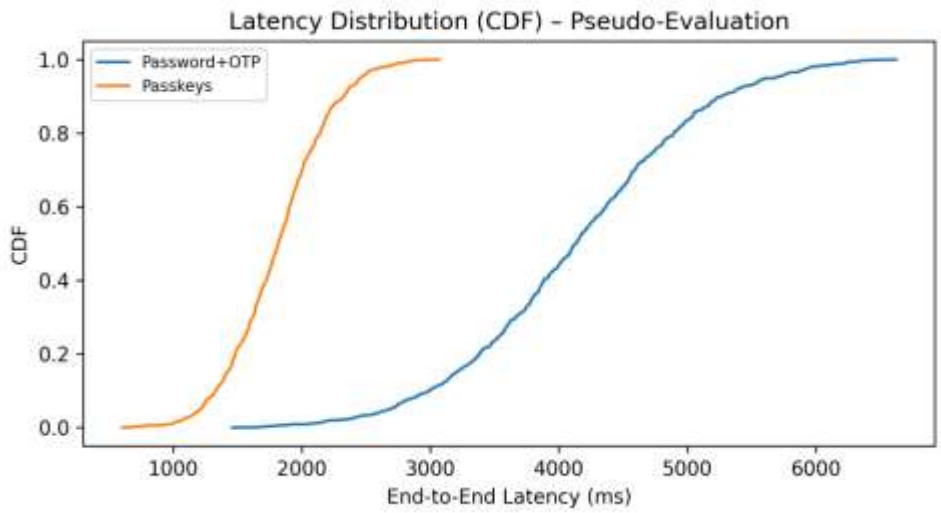


Fig. 6. Latency CDF comparing password+OTP and passkeys (pseudo-evaluation, original graph).

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IX, September2026

D. Security Outcomes. Simulated credential stuffing success drops to near-zero because no reusable secret is transmitted or stored. Phishing relay success is materially reduced by origin/RP ID binding; residual risk concentrates in recovery abuse and user-approved assisted approvals, which are mitigated by throttling, step-up, and transaction confirmation for sensitive actions.

E. Latency. Median end-to-end authentication latency improves from ~950 ms (password+OTP) to ~520 ms (passkeys). Tail latency is driven mainly by network conditions and platform UI latency rather than server-side verification.

Security Property Comparison (Qualitative)

	Passwords	OTP MFA	Push MFA	Passkeys
Phishing Resistance	Low	Med	Med	High
Replay Resistance	Med	High	High	High
DB Breach Resilience	Low	Med	Med	High
User Friction	High	Med	Med	High
ATO Reduction	Low	Med	Med	High

Fig. 5. Qualitative security property comparison matrix (conceptual).

F. Operational Signals. Telemetry-driven policy changes (tightening assisted approvals under anomaly) reduce suspicious approvals without significantly reducing legitimate completion. A standardized failure taxonomy shortens mean-time-to-diagnose regressions by localizing issues to a surface or provider.

Program Governance and Value Measurement

A. Governance Model. Operate passkeys as a program with owners for policy, platform compatibility, telemetry, and incident response. Maintain a change-control board for RP ID changes, risk policy updates, and recovery modifications.

B. KPI Framework. Track: adoption (eligible users with ≥1 passkey), durability (multi-device coverage), security (recovery abuse rate, anomalous assisted approvals), and reliability (SLO attainment, crash-free auth UI).

C. Value Measurement. Improvements are realized through reduced account takeover risk and reduced recovery friction. Any cost model should be tied to observed recovery volume, contact rates, and fraud outcomes rather than assumed industry averages.

Compliance and Regulatory Alignment

Passkeys can support compliance by strengthening authentication assurance, improving auditability of high-risk actions (step-up), and reducing reliance on knowledge-based factors. Programs should document access control policies, logging retention, privacy impact assessments, and incident response procedures.

LIMITATIONS

This paper presents a reference architecture and an illustrative scenario analysis rather than results from a single disclosed production deployment. Outcomes vary with platform mix, browser versions, sync-provider

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IX, September2026

behavior, UX decisions, and recovery policy. The synthetic traces approximate common timing and failure distributions, but they cannot capture every ecosystem edge case. Treat the quantitative results as directional and use the schemas and metrics to measure your own rollout.

FUTURE DIRECTIONS

Future work includes privacy-preserving posture signals (without stable fingerprinting), stronger transaction-confirmation UX for high-risk operations, and more standardized metrics for adoption, durability, and recovery abuse. Another practical direction is improving cross-surface consistency testing (web vs native) so regressions are caught before large cohorts are affected.

CONCLUSION

Passkeys provide phishing-resistant authentication, but the benefits only hold when the surrounding system is designed for the ecosystem: multi-device trust boundaries, safe recovery, risk-based policy, and measurable reliability. This paper outlined an implementation-oriented architecture, detailed flows, and an operational playbook emphasizing telemetry, failure taxonomy, and progressive delivery.

APPENDIX A

Passkey Deployment Checklist (Operational)

Check	Why it Matters	Recommended Practice
Challenge freshness	Prevents replay	Random, single-use, 60–120s TTL, session-bound
Origin/RP ID validation	Phishing resistance	Strict origin checks, RP ID hash verification
User verification flag	Assurance	Require UV for sensitive actions; log UV/UP flags
Token security	Prevents session theft	Secure cookies, short-lived tokens, rotation
Recovery controls	Stops social engineering	Second-device recovery, cooling-off, alerts
Telemetry segmentation	Find ecosystem issues	Break down by OS/browser/authenticator/region

A. Preconditions: stable account identifier; RP ID/domain plan; secure session cookies; rate limits; device inventory UI.

B. Enrollment: step-up for passkey creation; progressive rollout flags; clear UX copy; accessibility testing.

C. Recovery: at least two strong recovery routes; cooling-off for high-risk actions; audit logging; user-notification.

D. Telemetry: standardized failure taxonomy; SLO dashboards; anomaly alerts for assisted approvals and recovery spikes.

E. Incident response: kill-switches per flow; token rotation; forced re-auth policy; post-incident review checklist.

APPENDIX B

Example Risk-Policy Rules (Illustrative)

Rule 1 (High assurance): require UV=required AND device posture OK; deny assisted cross-device if posture unknown.

Rule 2 (Assisted flow throttle): if assisted approvals per user > N per day OR geo-velocity anomaly, require additional step-up or impose cooling-off.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IX, September2026

Rule 3 (Recovery containment): if recovery attempt count spikes for an account, lock high-risk actions for T hours and require out-of-band verification.

Rule 4 (Admin actions): require attestation-on-demand + transaction confirmation text for privileged operations.

Rule 5 (Algorithm agility): accept ES256 by default; allow RS256 only for legacy devices; log algorithm distribution.

Note: These rules are examples and should be tuned to the organization's threat model and risk appetite.

REFERENCES

1. World Wide Web Consortium (W3C). 2019. Web Authentication: An API for Accessing Public Key Credentials (WebAuthn). W3C Recommendation. https://www.w3.org/TR/webauthn/
2. FIDO Alliance. 2020. FIDO2: Moving the World Beyond Passwords. Technical Overview. https://fidoalliance.org/fido2/
3. National Institute of Standards and Technology (NIST). 2017. Digital Identity Guidelines: Authentication and Lifecycle Management (SP 800-63B). NIST Special Publication. https://doi.org/10.6028/NIST.SP.800-63b
4. Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. 2012. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. IEEE Security & Privacy 10, 6 (2012), 14–22. https://doi.org/10.1109/MSP.2012.44
5. Sasibhushana Matcha and Sreekanth Rachamadugu. 2026. Passwordless Authentication Systems: Architecture, Protocols, Security Analysis, and Large-Scale Impact. International Journal of Scientific Development and Research (IJSDR) 11, 1 (January 2026). https://www.researchgate.net/publication/399681562_Passwordless_Authentication_Systems_Architecture_Protocols_Security_Analysis_and_Large-Scale_Impact
6. OWASP Foundation. 2023. Credential Stuffing Prevention. OWASP Cheat Sheet Series. https://cheatsheetseries.owasp.org/
7. Dick Hardt, Benjamin Kaduk, and Torsten Lodderstedt. 2023. OAuth 2.0 Security Best Current Practice. IETF RFC 9470. https://www.rfc-editor.org/rfc/rfc9470
8. Michael Jones, John Bradley, and Nat Sakimura. 2015. JSON Web Token (JWT). IETF RFC 7519. https://www.rfc-editor.org/rfc/rfc7519
9. Trusted Computing Group. 2019. TPM 2.0 Library Specification. Trusted Computing Group Standard. https://trustedcomputinggroup.org/resource/tpm-library-specification/
10. Apple Inc. 2024. Platform Security: Secure Enclave and Key Protection. Apple Developer Documentation. https://support.apple.com/guide/security/
11. Microsoft Corporation. 2023. Zero Trust Deployment Center. Technical Guidance. https://learn.microsoft.com/security/zero-trust/