NG-KEY

Secure identity, readers, credentials, and audit visibility in one platform.

SECURITY PDF

SECURITY WHITEPAPER

NG-Key security whitepaper

A structured security view for NG-Key access control across platform security, strong authentication, credential media, deployment quality pipelines, ledger-backed traceability, and segmented runtime boundaries.

COVERAGE

- Platform security and least-privilege administration
- FIDO2, WebAuthn, and credential-media controls
- Deployment quality pipelines and protected rollout
- Auditability, ledger evidence, and segmented runtime boundaries

KEY FACTS

- Passkeys with FIDO2 and WebAuthn capable strong authentication
- Encrypted storage for sensitive secrets and keys
- Validated Git, CI/CD, PHPStan, and test quality gates
- Traceable access, provisioning, and commercial order events

SECURITY POSTURE

Access control is only as strong as the platform security beneath it, from authentication and transport to deployment quality and the final access result.

NG-Key is built for access control, so the security model begins with platform security: strong user authentication, protected credential handling, encrypted and integrity-protected communication paths, verifiable access outcomes, tightly controlled privileged administration, and disciplined production delivery.

The goal is not a single isolated control. The goal is a protected operating chain that reduces blind spots between the user ceremony, backend decisioning, runtime communication, release pipeline, and audit trail.

CORE SECURITY PILLARS

The security model combines platform security, stronger identity, controlled credential media, deployment quality, and traceable outcomes.

Platform security foundation

Privileged administration, secret handling, protected transport, runtime segmentation, and operational visibility provide the baseline on which secure access control depends.

Identity and credential strength

Passkeys with FIDO2 and WebAuthn capable flows support stronger authentication than reusable shared secrets and help bind access to real user devices.

Credential media and reader technologies

NG-Key supports deployments that combine modern public-key credentials with reader hardware for LEGIC and MIFARE-class environments, and the current credential path can carry DESFire-linked identifiers where physical media is part of the rollout.

Ledger and verification trail

Canonical events, hashes, signatures, and blockchain-backed verification trails strengthen the inspectability of access audits, permission changes, and related high-impact events.

IMPORTANT SECURITY DOMAINS

Key controls and design principles

Identity and authentication

Use passkeys and FIDO2 or WebAuthn capable flows, verified enrollment paths, and explicit account ownership to reduce weak authentication exposure.

Encryption and transport protection

Protect traffic with HTTPS, TLS, or equivalent encrypted transport controls, keep stored secrets encrypted at rest, and keep end-to-end encryption enabled where the dependent path supports it.

Deployment quality and production rollout

Stronger production security starts before runtime: validated Git workflows, CI or CD quality gates, formatting, static analysis, tests, and controlled promotion into immutable runtime images reduce avoidable rollout risk.

Audit, ledger, and commercial traceability

Keep access events, provisioning changes, privileged actions, and order-related handover or fulfillment events traceable so investigations and compliance reviews have usable context across operational and commercial flows.

ADDITIONAL IMPORTANT POINTS

- Where a dependent client, reader, or integration path supports end-to-end encryption, it should remain enabled and verifiable. Where that is not available, each hop should still be protected with HTTPS, TLS, or equivalent encrypted transport controls.
- Restrict identity integration settings, broker credentials, and reader-specific secrets to authorized roles and review them periodically, and keep sensitive settings encrypted at rest.
- Keep production secrets on the target hosts or equivalent protected secret stores and out of Git history, exports, and screenshots.
- Use delivery pipelines that verify formatting, builds, static analysis, and tests before production deployment so quality failures are caught before they become security incidents.
- Review monitoring, alerts, ledger failures, and operational signals as part of security operations because degraded runtime behavior can become a security event.
- Use explicit ownership, invitation hygiene, disciplined offboarding, and reviewable card or transponder assignment flows so dormant privileged or credential access does not accumulate over time.

SHARED RESPONSIBILITY

Security remains strongest when platform controls, delivery discipline, and customer operations work together.

NG-Key provides platform controls for strong authentication, encrypted secret storage, protected transport options, credential and media handling, audit visibility, ledger-backed evidence, and traceable access decisions. Customers and partners still need disciplined identity governance, endpoint hygiene, approval workflows, ordering controls, and operational review around who receives access and who keeps administrative responsibility.

That shared-responsibility model is strongest when it stays explicit: technical safeguards, validated delivery, runtime visibility, and accountable operating practice reinforce each other instead of leaving gaps between teams.
