RFC-1300: Device Authority & Operating System Integration Model

Title: Device Authority & Operating System Integration Model

Status: Draft

Category: Standards Track

Version: 1.0

Date: March 2026

Authors: Swoop Engineering Team

Canonical URL: /rfc-1300

Abstract

This specification defines the cryptographic and architectural requirements for establishing **device-bound authority** within the UniKey ecosystem. By leveraging hardware-backed security boundaries and operating system integrity, RFC-1300 ensures that authorization originates from a specific physical endpoint rather than reusable digital credentials. This model establishes the **"Hardware Root of Trust"** necessary for the secure generation of Trust Packets, enabling high-assurance actions in distributed, agentic environments while materially mitigating the risk of scalable remote fraud.

---

1. Purpose

This specification defines how UniKey establishes **device-bound cryptographic authority** and how operating system security boundaries protect that authority.

The purpose of this document is to describe the security properties required for devices that generate Trust Packets and authorize digital actions.

Device authority ensures that authorization originates from a specific device security boundary rather than from reusable credentials such as passwords, tokens, or API keys.

This specification defines:

- the device authority model
- requirements for device key generation and protection
- operating system security assumptions
- authorization interaction requirements
- integration expectations for device platforms

This specification does not mandate a specific hardware platform, biometric mechanism, or operating system vendor.

---

2. Security Objective

The security objective of device authority is to ensure that the authorization of a digital action requires **possession of a specific device and access to its protected signing key**.

Under this model:

Credential compromise alone MUST NOT allow execution of actions.

Authorization requires the following:

1. possession of the device
2. access to the device’s private signing key
3. explicit authorization by the device user or device policy

This model materially reduces the feasibility of scalable remote fraud.

---

3. Device Authority Model

In the UniKey architecture, the device acts as a **cryptographic authority endpoint** capable of issuing signatures over Trust Packets.

The device authority model consists of the following elements:

Device Private Key
A cryptographic signing key stored within a protected device security boundary.

Device Identity
 An identifier associated with the device key.

Authorization Event
 A local event where the device confirms approval for a specific action.

Trust Packet Signature
 The device signs the canonical Trust Packet Hash (TPH) defined in RFC-001.

The resulting Trust Packet contains verifiable proof that the device approved the action.

---

4. Device Key Generation and Storage

Each participating device MUST generate or obtain a cryptographic signing key pair.

Device private keys MUST be protected by the device security boundary.

Recommended implementations include:

• secure enclave hardware
 • trusted execution environments
 • hardware-backed keystores

Private keys MUST NOT be exportable in plaintext form.

Public keys MUST be discoverable by verifiers using distributed key infrastructure mechanisms defined by UniKey.

---

5. Device Authorization Interaction

Before signing a Trust Packet for a discrete action, the device MUST confirm authorization according to device policy.

Authorization confirmation MAY include:

• biometric authentication (e.g., Face ID, fingerprint)
 • device passcode entry
 • explicit user confirmation via system prompt
 • trusted application policy

The authorization prompt SHOULD clearly display the action context derived from the Trust Packet payload.

Examples of context displayed:

• transaction amount
• merchant or relying party
• requested action
• destination or target

This ensures that user approval corresponds to the actual action being authorized.

---

6. Device Signing Process

When a device approves an action, it performs the following steps:

1. Receive the Trust Packet canonical representation (excluding signatures).
2. Verify the action context presented to the user.
3. Generate the Trust Packet Hash (TPH) as defined in RFC-001.
4. Sign the TPH using the device private key.
5. Return the signature as part of the Trust Packet signatures array.

The device signature binds the following fields:

• action payload
• audience
• context
• nonce
• expiration

This ensures that the authorization is specific, fresh, and non-replayable.

---

7. Operating System Security Boundary

The UniKey device authority model assumes that the operating system enforces a security boundary protecting:

- private key storage
- biometric authentication mechanisms
- authorization prompts
- signing operations

The operating system MUST prevent untrusted applications from extracting private keys or silently triggering signatures.

Examples of platform protections include:

- application sandboxing
- secure key storage
- trusted user interface prompts
- secure enclave or hardware-backed key management

UniKey implementations rely on the integrity of this boundary.

---

8. Malware and Device Compromise Considerations

Device compromise remains a possible attack scenario.

Examples include:

- malware with full system privileges
- compromised operating systems
- stolen unlocked devices

UniKey does not eliminate these attacks.

However, the architecture shifts fraud from **scalable remote credential abuse** to attacks requiring **device compromise or user deception**, which are significantly more difficult to automate at scale.

---

9. Delegated Device Authority

Devices MAY delegate limited authority to trusted agents or applications.

Delegation MUST follow the Trust Packet delegation rules defined in RFC-001.

Delegated packets MUST:

• reference the parent Trust Packet hash
• narrow the permitted scope
• maintain the same audience and context constraints
• include an independent signature from the delegated authority

Delegation MUST NOT expand authority.

---

10. Device Registration and Key Discovery

Verifiers must be able to obtain the public key associated with a device authority.

Public key discovery may occur through:

• domain-based key infrastructure (DNS/DKIM distribution)
• registry-based discovery
• platform key directories

The mechanism used MUST allow verifiers to validate the device signature independently.

The security of the Trust Packet does not rely on a centralized identity provider.

---

11. Security Properties

When implemented according to this specification and RFC-001, device authority provides the following properties:

Device Possession Requirement
Authorization requires the signing device.

Credential Theft Resistance
Stolen passwords, tokens, or account credentials alone cannot authorize actions.

Action Binding
The device signature is bound to a specific action payload.

Audience Binding
Authorization is valid only for the intended verifier.

Replay Resistance
Authorization includes freshness constraints enforced by verifiers.

12. Relationship to Other Security Models

The UniKey device authority model is consistent with security assumptions used by modern device-bound authentication systems such as hardware security keys, FIDO passkeys, and device-based payment authorization systems.

These systems rely on the device security boundary to protect private signing keys.

UniKey extends this model to authorize arbitrary digital actions across distributed systems.

---

13. Implementation Independence

This specification defines required security properties but does not mandate specific hardware, operating systems, or biometric technologies.

Implementations may vary across device platforms provided they maintain the required security invariants.

---

14. Conformance

A device implementation is conformant with this specification if it:

- generates or securely stores a private signing key
- performs user authorization confirmation before signing
- signs the Trust Packet Hash as defined in RFC-001
- protects the private key within the device security boundary
- prevents silent signing by untrusted software

Devices failing these requirements MUST NOT be treated as trusted authorities.