Network Working Group
Internet-Draft
Intended status: Informational
Expires: 22 July 2026

X. Zinn
Independent

18 January 2026

SoftHSM Enforcement Rules
draft-xzinn-softhsm-enforcement-rules-00

Abstract

This document defines enforcement and governance rules for the SoftHSM trust model.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 July 2026.

Table of Contents

1. Introduction

Abstract

This document defines enforcement rules for software based hardware security modules used in identity based systems. The rules specify what a SoftHSM MUST enforce, what it MUST ignore, and what it MUST never infer. By constraining signing behavior to explicit, verifiable inputs, the model ensures deterministic authorization, prevents semantic confusion, and eliminates classes of attacks caused by implicit trust or contextual inference.

Introduction

Cryptographic signing environments are frequently treated as passive tools that perform requested operations without evaluating intent, scope, or context. This approach places responsibility for correctness entirely on calling applications and exposes signing keys to misuse, ambiguity, or unintended authorization.

This document defines a SoftHSM enforcement model in which the signing environment actively enforces explicit rules before performing any cryptographic operation. The SoftHSM does not interpret payloads, infer intent, or apply heuristics. It evaluates only declared, verifiable inputs and enforces deterministic constraints.

The SoftHSM is positioned as the final authority prior to cryptographic signing. It is the last opportunity to prevent misuse of keys, mis binding of semantics, or execution of unauthorized actions.

This specification applies to software based HSM implementations operating on general purpose systems. It does not define cryptographic algorithms, key storage formats, or physical security properties. Instead, it defines behavioral constraints that any compliant SoftHSM MUST implement regardless of underlying cryptographic primitives.

This document builds on the message envelope format, purpose semantics, consent model, and capability token mechanisms defined in earlier specifications. It defines how those inputs are enforced at the point where signatures are created.

2. Scope and Goals

By making enforcement explicit and non inferential, the SoftHSM becomes a deterministic policy boundary rather than a passive cryptographic service.

Terminology

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in RFC 2119 and RFC 8174.

SoftHSM A software based hardware security module that performs cryptographic operations while enforcing explicit policy and behavioral constraints.

Signing Operation A cryptographic operation that produces a signature over defined input data using a private key managed by the SoftHSM.

Signing Request A structured request presented to the SoftHSM that specifies the data to be signed and the context under which signing is requested.

Required Inputs The explicit, verifiable inputs that MUST be present in a signing request for the SoftHSM to consider performing a signing operation.

Enforcement The act of evaluating a signing request against defined rules and constraints prior to executing a cryptographic operation.

Deterministic Decision A signing decision that is fully determined by explicit inputs and configuration, producing the same outcome given the same inputs.

Inference Any attempt to derive intent, authorization, or semantics from context, payload content, transport, or external state not explicitly provided in the signing request.

Policy Local rules that constrain signing behavior, including which keys may be used, under what conditions, and for which purposes.

Audit Record A verifiable record of a signing decision, including inputs evaluated and the resulting outcome.

Key Material Cryptographic keys managed by the SoftHSM, including private keys and associated metadata.

Signing Context The declared purpose, identity, and scope associated with a signing request.

3. Threat Model

Execution Any action taken as a result of a signing operation beyond the production of a cryptographic signature. Execution semantics are explicitly outside the scope of the SoftHSM.

Enforcement Model

The enforcement model defines how a SoftHSM evaluates signing requests and determines whether a cryptographic operation is permitted.

The SoftHSM acts as a policy enforcement point. It evaluates explicit inputs presented in a signing request against local policy and enforcement rules before performing any cryptographic operation.

The SoftHSM MUST operate deterministically. Given the same signing request, policy configuration, and key state, the SoftHSM MUST reach the same decision and produce the same outcome.

The SoftHSM evaluates only explicit inputs. It MUST NOT inspect payload contents, interpret application semantics, infer intent, or consult external context not explicitly provided in the signing request.

The enforcement model follows a fixed evaluation order.

First, the SoftHSM MUST validate the structure and completeness of the signing request. Second, the SoftHSM MUST verify the authenticity and integrity of the signing request inputs. Third, the SoftHSM MUST evaluate required inputs against local policy. Fourth, the SoftHSM MUST determine whether the requested signing operation is permitted.

If any step fails, the signing operation MUST be denied.

The SoftHSM MUST fail closed. Absence of required inputs, ambiguity, or policy mismatch MUST result in denial rather than permissive behavior.

The SoftHSM MUST treat signing requests as untrusted input. Even requests originating from local applications or privileged processes MUST be evaluated identically.

The enforcement model intentionally separates authorization from cryptographic execution. Authorization is determined entirely before any cryptographic operation occurs.

By constraining evaluation to explicit, verifiable inputs and enforcing a fixed decision order, the enforcement model prevents misuse of signing keys and eliminates classes of attacks based on implicit trust or contextual inference.

The SoftHSM may support cryptographic operations other than signing, such as encryption or decryption. However, this specification defines enforcement rules only for signing operations. Requirements for encryption, key agreement, or confidentiality services are explicitly out of scope and may be defined by separate specifications.

4. Enforcement Principles

Any enforcement failure MUST result in deterministic failure and MUST NOT degrade into warnings, partial success, fallback behavior, or best effort execution.

Identity Bound Decryption Enforcement

This section defines enforcement rules that bind decryption authority to a specific SoftHSM instance.

A SoftHSM MUST possess a unique internal cryptographic identity for the purpose of enforcement.

When identity bound encryption is applied to an artifact, the SoftHSM MUST bind the encryption operation to its own SoftHSM identity.

A SoftHSM MUST NOT permit decryption of an identity bound artifact unless the decryption operation is executed within the same authorized SoftHSM identity.

Possession of valid cryptographic keys alone MUST NOT be sufficient to permit decryption.

If an encrypted artifact is copied, exfiltrated, cloned, or otherwise removed from the authorized SoftHSM execution environment, decryption MUST fail deterministically.

A SoftHSM MUST enforce this restriction regardless of key availability, key correctness, or external system state.

Transfer of decryption authority to a different SoftHSM instance MUST require an explicit governance action as defined by the governance rules governing the SoftHSM environment.

Silent key re wrapping, implicit trust inheritance, or undeclared SoftHSM substitution MUST NOT be permitted.

A SoftHSM MUST record its identity as part of the audit record for any encryption or decryption operation involving identity bound artifacts.

Required Inputs

A SoftHSM MUST require a complete and explicit set of inputs for every signing request. Absence, ambiguity, or partial specification of required inputs MUST result in denial.

5. Capability Enforcement

The SoftHSM MUST NOT assume defaults, infer missing values, or derive intent from context. Only explicitly provided inputs are authoritative.

At a minimum, a signing request MUST include the following required inputs.

Signing Key Identifier An explicit identifier for the private key requested for use. The SoftHSM MUST NOT select keys implicitly or automatically.

Data to be Signed The exact data or cryptographic digest to be signed. The SoftHSM MUST treat this data as opaque and MUST NOT inspect or interpret its contents.

Declared Purpose The purpose associated with the signing request. The purpose MUST be explicit and MUST correspond to a supported purpose value. Signing requests without a declared purpose are invalid.

Requesting Identity The identity on whose behalf the signing request is made. The SoftHSM MUST verify that this identity is authorized to request use of the specified key.

Scope and Context Explicit scope information required to evaluate authorization, such as direction, resource class, or capability reference, when applicable.

Authorization Evidence Verifiable evidence required by local policy to authorize signing, such as consent state, capability token reference, or policy identifier. The SoftHSM MUST NOT attempt to retrieve or infer such evidence implicitly.

Request Integrity A mechanism to ensure the integrity and authenticity of the signing request itself. The SoftHSM MUST reject requests that cannot be verified as intact and authentic.

All required inputs MUST be bound together as part of the signing request. Modification of any required input after request submission MUST invalidate the request.

The SoftHSM MUST validate required inputs prior to any policy evaluation or cryptographic operation. Requests that fail validation MUST be denied without side effects.

By requiring explicit and complete inputs, the SoftHSM ensures that signing decisions are intentional, reviewable, and immune to contextual ambiguity.

Prohibited Inference

The SoftHSM MUST NOT infer intent, authorization, or semantics beyond what is explicitly provided in a signing request.

6. Governance Enforcement

Inference is prohibited because it introduces ambiguity, non determinism, and hidden policy decisions that undermine the security guarantees of the signing environment.

The SoftHSM MUST NOT infer or derive meaning from any of the following.

Payload content Transport protocol or source Application identity or process context Historical behavior or prior requests User interface state or presentation Time of day or execution environment Absence of required inputs

The SoftHSM MUST NOT attempt to classify, interpret, or inspect the data being signed. The data to be signed is treated as opaque input.

The SoftHSM MUST NOT broaden scope, escalate privilege, or relax constraints based on perceived legitimacy, repetition, or convenience.

The SoftHSM MUST NOT attempt to compensate for incomplete requests. Missing or ambiguous inputs MUST result in denial rather than fallback behavior.

The SoftHSM MUST NOT infer authorization from identity recognition alone. Possession of a key or identity association does not imply permission to sign.

Any behavior that results in different signing decisions for equivalent explicit inputs is prohibited.

By explicitly forbidding inference, the SoftHSM enforces a strict separation between policy, authorization, and cryptographic execution. This ensures that signing decisions remain deterministic, auditable, and resistant to misuse.

Signing Decision Rules

Signing decision rules define how the SoftHSM determines whether to perform a signing operation once required inputs have been validated.

A signing operation MUST be permitted only if all of the following conditions are satisfied.

’ All required inputs are present, well formed, and authenticated ’ The requested signing key exists and is available for use ’ The requesting identity is authorized to use the specified key ’ The declared purpose is supported and permitted for the key ’ Scope and context constraints are satisfied ’ Required authorization evidence is present and valid ’ No revocation or policy restriction applies

If any condition fails, the signing operation MUST be denied.

7. Operational Constraints

The SoftHSM MUST evaluate signing requests using a fixed and documented decision order. Evaluation order MUST NOT vary based on request content or origin.

The SoftHSM MUST NOT perform partial signing or conditional signing. Signing is an atomic operation that either succeeds completely or fails without side effects.

The SoftHSM MUST NOT modify, normalize, or reinterpret the data being signed. The signature MUST apply exactly to the data presented in the request.

The SoftHSM MUST NOT expose private key material or intermediate cryptographic state as part of a signing decision.

The SoftHSM MUST return a clear and deterministic outcome for each signing request, indicating approval or denial. Error reporting MUST NOT leak sensitive policy or key information.

By enforcing explicit and ordered decision rules, the SoftHSM ensures that cryptographic operations occur only when authorization is unambiguous and complete.

Audit and Determinism

Audit and determinism requirements ensure that SoftHSM behavior is observable, reproducible, and resistant to silent policy drift.

The SoftHSM MUST produce deterministic outcomes. Given the same signing request, policy configuration, and key state, the SoftHSM MUST reach the same decision and produce identical results.

The SoftHSM SHOULD record an audit record for each signing request. Audit records SHOULD include sufficient information to reconstruct the decision without exposing sensitive key material or payload contents.

At a minimum, audit records SHOULD capture:

’ Timestamp of the request ’ Signing key identifier ’ Requesting identity ’ Declared purpose ’ Decision outcome ’ Policy reference applied

Audit records MUST NOT include private key material or raw payload data. Where payload digests are recorded, they MUST be treated as sensitive metadata.

Audit mechanisms MUST NOT influence signing decisions. Failure to record an audit event MUST NOT cause a signing operation to succeed or fail differently.

8. Failure Modes

Audit records SHOULD be protected against modification and unauthorized access. Integrity of audit records is essential for post hoc verification and dispute resolution.

The SoftHSM MUST NOT rely on audit state to make authorization decisions. Audit is observational, not operational.

By enforcing determinism and producing verifiable audit trails, the SoftHSM enables accountability without introducing inference or coupling between authorization and observation.

Security Considerations

The SoftHSM represents the final authorization boundary prior to cryptographic signing and therefore must be treated as a high value security component.

The primary security objective of the SoftHSM enforcement model is to prevent unauthorized or unintended use of signing keys. By requiring explicit inputs and prohibiting inference, the model reduces the risk of confused deputy attacks, privilege escalation, and semantic misbinding.

Key misuse is a central threat. The SoftHSM MUST ensure that keys are used only for explicitly authorized purposes and scopes. Failure to enforce purpose or scope constraints may result in signatures that assert unintended meaning or authority.

Signing request forgery is another risk. The SoftHSM MUST verify the integrity and authenticity of signing requests and MUST reject requests that cannot be validated.

Policy misconfiguration may lead to overly permissive behavior. Implementations SHOULD provide mechanisms to validate policy consistency and SHOULD favor fail closed behavior in the presence of ambiguity.

Side channel leakage, such as timing differences or error messages, may reveal information about policy or key availability. Implementations SHOULD minimize observable differences between denial reasons and SHOULD avoid exposing detailed failure modes to untrusted callers.

Compromise of the host system threatens the SoftHSM. While this document does not define host hardening requirements, implementations SHOULD consider isolation mechanisms, access controls, and monitoring appropriate to their threat model.

By enforcing deterministic authorization and strict separation of concerns, the SoftHSM reduces reliance on application correctness and provides a robust defense against classes of attacks caused by implicit trust or contextual interpretation.

Security Considerations

The SoftHSM represents the final authorization boundary prior to cryptographic signing and therefore must be treated as a high value security component.

9. Security Considerations

The primary security objective of the SoftHSM enforcement model is to prevent unauthorized or unintended use of signing keys. By requiring explicit inputs and prohibiting inference, the model reduces the risk of confused deputy attacks, privilege escalation, and semantic misbinding.

Key misuse is a central threat. The SoftHSM MUST ensure that keys are used only for explicitly authorized purposes and scopes. Failure to enforce purpose or scope constraints may result in signatures that assert unintended meaning or authority.

Signing request forgery is another risk. The SoftHSM MUST verify the integrity and authenticity of signing requests and MUST reject requests that cannot be validated.

Policy misconfiguration may lead to overly permissive behavior. Implementations SHOULD provide mechanisms to validate policy consistency and SHOULD favor fail closed behavior in the presence of ambiguity.

Side channel leakage, such as timing differences or error messages, may reveal information about policy or key availability. Implementations SHOULD minimize observable differences between denial reasons and SHOULD avoid exposing detailed failure modes to untrusted callers.

Compromise of the host system threatens the SoftHSM. While this document does not define host hardening requirements, implementations SHOULD consider isolation mechanisms, access controls, and monitoring appropriate to their threat model.

By enforcing deterministic authorization and strict separation of concerns, the SoftHSM reduces reliance on application correctness and provides a robust defense against classes of attacks caused by implicit trust or contextual interpretation.

IANA Considerations

This document does not define any new namespaces, registries, protocol numbers, or parameters requiring action by IANA.

If future specifications define registries related to SoftHSM policy identifiers or standardized signing contexts, those documents will specify their own IANA considerations.

References

Normative References

<<RFC2119>> Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

<<RFC8174>> Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017.

Informative References

<<FIPS140-3>> National Institute of Standards and Technology, "Security Requirements for Cryptographic Modules", FIPS PUB 140-3, March 2019.