RFC 003 DKIM / DNS Key Discovery & Validation Profile

Status: Draft (Normative)

Purpose: Define a global, deterministic profile for cryptographic key discovery, validation, and trust using DKIM and DNS as the public trust fabric.

This document specifies how UniKey implementations discover, validate, cache, rotate, and revoke public keys used to verify Trust Packets.

1. Overview

UniKey relies on existing, globally deployed DNS and DKIM infrastructure to distribute and validate public keys at internet scale.

This profile:

- Treats DNS as a **public key directory**, not a message transport
- Treats DKIM as a **signature container**, not an email feature
- Defines stricter validation and operational rules than baseline DKIM

The objective is deterministic, infrastructure-level trust that does not require bilateral integration, accounts, or shared state.

2. Key Roles

2.1 Issuer

The **Issuer** is the root authority asserting identity and authority.

Responsibilities:

- Publishes public keys via DNS
- Signs Trust Packets with corresponding private keys
- Maintains rotation and revocation discipline

Examples:

- Merchant domain
- Device manufacturer

- Payment network

2.2 Delegate

A **Delegate** is an entity granted limited authority by an Issuer.

Characteristics:

- Operates under explicitly delegated scope
- Signs Trust Packets with its own private key
- Must reference the Issuer’s delegation packet

Delegation MUST narrow authority and MUST be independently verifiable.

2.3 Cross-Domain Delegation Example (Normative)

Scenario:

- `merchant.example` (Issuer) delegates payment authorization to `psp.example`
- `psp.example` signs a Trust Packet for a specific transaction

Flow:

1. Issuer publishes its DKIM public key via DNS
2. Issuer issues a delegation Trust Packet authorizing `psp.example` for limited scope
3. Delegate signs a transaction-specific Trust Packet
4. Verifier:
    - Validates delegate signature via DNS
    - Verifies delegation packet chain
    - Confirms scope reduction and validity

At no point does authority expand beyond what the Issuer granted.

2.1 Issuer

The **Issuer** is the root authority asserting identity and authority.

Responsibilities:

- Publishes public keys via DNS
- Signs Trust Packets with corresponding private keys
- Maintains rotation and revocation discipline

Examples:

- Merchant domain
- Device manufacturer
- Payment network

2.2 Delegate

A **Delegate** is an entity granted limited authority by an Issuer.

Characteristics:

- Operates under explicitly delegated scope
- Signs Trust Packets with its own private key
- Must reference the Issuer’s delegation packet

Delegation MUST narrow authority and MUST be independently verifiable.

3. DNS Record Formats

Public keys are published using DKIM-compatible DNS TXT records.

Format:

```
None
<selector>._domainkey.<domain> IN TXT "v=DKIM1; k=rsa;
p=<base64url>"
```

Requirements:

- Only DKIM-compatible formats are permitted
- Records MUST contain exactly one public key
- Unsupported or unknown tags MUST cause rejection

UniKey implementations MUST NOT rely on email headers, SMTP metadata, or message bodies for key discovery.

3A. Normative DKIM Tag Profile

This profile restricts DKIM tags to ensure deterministic verification.

Allowed Tags

- `v` — MUST be `DKIM1`
- `k` — Key type (e.g., `rsa`, `ec`)
- `p` — Base64url-encoded public key material

Conditionally Allowed Tags

- `t` — MAY be used for testing flags but MUST NOT weaken verification

Disallowed Tags

- `g` — Granular address matching (not applicable)
- `h` — Header field restrictions
- `n` — Notes or comments
- Any experimental or non-standard tags

Presence of disallowed tags MUST cause verification failure.

Public keys are published using DKIM-compatible DNS TXT records.

Format:

```
None
<selector>._domainkey.<domain> IN TXT "v=DKIM1; k=rsa;
p=<base64url>"
```

Requirements:

- Only DKIM-compatible formats are permitted
- Records MUST NOT contain multiple public keys
- Unsupported tags MUST be ignored

UniKey implementations MUST NOT rely on email headers, SMTP metadata, or message bodies for key discovery.

4. Selector Usage

Selectors identify specific public keys within a domain.

Rules:

- Selectors MUST be unique per active key
- Selectors SHOULD encode rotation epoch or key purpose
- Reuse of selectors for different keys is prohibited

Example:

```None
2026q1._domainkey.example.com
```

---

5. Key Algorithms & Minimum Strength

This profile defines minimum cryptographic requirements.

Algorithm	Minimum
RSA	2048-bit
ECC	P-256
Hash	SHA-256 or stronger

Verifiers MUST reject keys below minimum strength.

6. Caching & TTL Guidance

Verifiers MAY cache validated public keys.

Guidance:

- Honor DNS TTLs
- Enforce maximum cache lifetimes
- Bind cache entries to selector + domain

Recommended:

- Maximum cache lifetime: 24 hours
- Minimum revalidation: on TTL expiry

Caching MUST NOT bypass revocation or compromise detection.

7. Key Rotation & Revocation

7.1 Rotation

Issuers SHOULD rotate keys regularly.

Rules:

- New selectors MUST be published before use
- Old keys MUST remain available through overlap window
- Overlap window SHOULD be ≥ DNS TTL

7.2 Revocation

Revocation is achieved by:

- Removing DNS records
- Setting `p=` to empty value

Verifiers MUST treat missing or empty keys as invalid.

7A. Key Compromise Scenarios

If a private key is suspected or confirmed compromised:

1. The Issuer MUST immediately revoke the affected key by removing or nullifying the DNS record
2. A new key pair MUST be generated and published under a new selector
3. All Trust Packets signed after compromise detection MUST use the new selector
4. Verifiers MUST reject packets signed with revoked selectors

Implementations MAY apply additional scrutiny or temporary rejection to packets signed shortly before revocation.

7.1 Rotation

Issuers SHOULD rotate keys regularly.

Rules:

- New selectors MUST be published before use
- Old keys MUST remain available through overlap window
- Overlap window SHOULD be ≥ DNS TTL

7.2 Revocation

Revocation is achieved by:

- Removing DNS records
- Setting `p=` to empty value

Verifiers MUST treat missing or empty keys as invalid.

8. DNS Hardening Profile

UniKey implementations MUST apply additional DNS validation beyond baseline DKIM.

Required measures:

- Multi-resolver validation
- Change detection on key material
- Anomaly detection on TTL or record structure

DNSSEC MAY be used but MUST NOT be solely relied upon.

9. Failure & Fallback Behavior

Failure conditions:

- Key not found
- Key malformed
- Signature mismatch
- DNS anomalies detected

Behavior:

- Fail closed for all authentication decisions
- No silent downgrade to weaker mechanisms
- Fallback MAY only occur if explicitly authorized by policy

10. Explicit Contrast with Baseline DKIM

This profile differs from baseline DKIM in the following ways:

- DKIM is used exclusively for public key discovery and signature validation, not for email authentication
- Only a strict subset of DKIM tags is permitted
- Keys are treated as global identity anchors, not message-scoped artifacts
- DNS responses are actively validated, monitored, and correlated
- Verification is fail-closed and infrastructure-enforced

Baseline DKIM optimizes for mail delivery. This profile optimizes for deterministic trust.

11. Security Considerations

Threats addressed:

- DNS spoofing or poisoning
- Key substitution
- Replay with stale keys
- Delegation abuse

Mitigations:

- Deterministic DNS discovery
- Cryptographic verification
- Strict selector discipline
- Infrastructure-level enforcement

This profile treats DNS as a distributed, append-only trust directory whose integrity is continuously monitored.

12. Conformance

An implementation is conformant if it:

- Discovers keys using this profile
- Enforces minimum cryptographic requirements

- Rejects unverifiable or anomalous keys
- Applies DNS hardening checks

This document defines the global trust fabric underpinning UniKey verification.