Network Working Group
Internet-Draft
Intended status: Informational
Expires: 6 July 2026

B.A. Fisher
DPA R&D Ltd (https://www.dpa-cloud.co.uk)

2 January 2026

TLS-DPA: An Identity-Bound Security Protocol for Traditional, Overlay, and Zero-Port Transports
draft-dpa-tls-dpa-00

Abstract

TLS-DPA is an experimental, identity-bound security protocol inspired by the design of TLS 1.3 ([RFC8446]). It is intended to operate consistently across environments where conventional IP address and port semantics are weak, unstable, or intentionally absent, including zero-port transports such as UZP ([UZP]). TLS-DPA generalises the handshake so it is not tied to server-side listeners, binds authentication to Service Identities rather than network coordinates, reduces metadata exposure to intermediaries (including rendezvous nodes in UZP fabrics), provides a unified hybrid-KEM post-quantum transition model ([NIST-PQC]), and supports session continuity across overlay path changes (e.g., QUIC Connection IDs; [RFC9000]).

Note to Reviewers

This document is an Internet-Draft derived from internal research material solely to enable structured technical review, interoperability discussion, and disciplined specification development under the Internet-Draft process. It is a work-in-progress research artefact and does not constitute a standard, recommendation, or finished specification.

The name TLS-DPA is used to label this research protocol and avoid confusion with the IETF TLS versioning and registry space. It is not presented as a new version of the IETF TLS protocol, and no IANA allocations are requested by this draft.

Where this document provides numeric guidance (for example, replay windows, resumption behaviour, or profile parameters), the intent is to offer recommended bounds suitable for experimentation; profile-based behaviour and implementation discretion are explicitly expected within stated limits.

The text aims to preserve a clear separation of normative and informative material. Requirement words are used only where protocol behaviour is intentionally specified, and the draft avoids implying standards-track status or mandatory implementation.

This document is intended to support early peer review and international collaboration while retaining flexibility for substantial revision, experimental implementation, and validation. No patent grants or licensing commitments are implied beyond the IETF Trust provisions applicable to Internet-Drafts.

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 6 July 2026.

Copyright Notice

Table of Contents

1. Scope and Status

This Internet-Draft specifies TLS-DPA, an experimental security protocol intended for identity-first and topology-independent deployments, including rendezvous and zero-port fabrics. The goal is to support early review and implementation experiments; substantial revision is expected.

TLS-DPA is designed for environments where conventional port-listening assumptions and IP:port-based identity binding do not hold. It is not a universal replacement, is not mandated outside its target environment, and is designed for experimentation and profile-driven deployments.

2. Introduction

TLS 1.3 ([RFC8446]) defines the current baseline for transport-layer security on the Internet. However, its usage patterns remain oriented around server-side listeners bound to IP address and port tuples, and many deployments treat these network coordinates as meaningful anchors for authentication and policy.

TLS-DPA extends the design principles of TLS 1.3 to support:

* operation over identity-first, topology-independent transports (for example UZP; [UZP]);
* authentication bound to Service Identities, rather than IP addresses and ports;
* reduced metadata exposure to intermediaries, including rendezvous nodes in UZP fabrics;
* hybrid classical/post-quantum KEM negotiation aligned with the NIST PQC process ([NIST-PQC]);
* session continuity across transport or overlay path changes (for example QUIC Connection IDs; [RFC9000]).

The TLS-DPA wire image is intended to remain close to TLS 1.3, enabling reuse of existing implementation structure while adding explicit identity and transport binding into the handshake transcript and key schedule.

TLS-DPA also aligns with zero-trust guidance (NIST SP 800-207 [NIST-SP800-207]) and identity-centric designs such as HIP [RFC7401].

3. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

Terminology used throughout this document:

CID Canonical Identity (a long-term public key hash).

EID Ephemeral Identity (a session-level fingerprint).

UZP Zero-port transport as defined by the companion UZP Internet-

Draft ([UZP]).

ZPIT Zero-Port Interconnect Tunnel (a UZP fabric channel).

Pantheon A policy/identity authority providing session grants and capability metadata.

Service Identity The identity to which TLS-DPA authentication is bound (for example a DNS name, CID, EID, or a UZPIF selector).

4. Design Goals

TLS-DPA is designed to:

1. decouple channel authentication from IP address and port topology;

2. provide identity-first naming independent of network routing;

3. support hybrid classical and post-quantum KEM negotiation aligned with NIST PQC guidance ([NIST-PQC]);

4. reduce metadata in early handshake flights;

5. bind channels to transport-level identifiers (for example UZP SessionIDs or QUIC Connection IDs; [RFC9000]);

6. remain closely aligned with the structure of TLS 1.3 ([RFC8446]);

7. operate efficiently over UZP and UZPIF rendezvous fabrics ([UZP] , [UZPIF]).

Where this document specifies algorithms or parameter sets (for example hybrid KEM combinations), these are intended as recommended profiles and may evolve. Implementations may support additional profiles and apply implementation-defined choices within any explicit limits described in the relevant sections.

5. Overview of TLS-DPA

TLS-DPA retains the basic architecture of TLS 1.3 ([RFC8446]) but introduces:

* transport-agnostic channel binding, via a dedicated extension that carries a transport identifier and class;

* Service Identity negotiation and binding into the transcript;

* mandatory transcript binding of identity and transport metadata;

* PQ-ready hybrid KEM negotiation using an explicit parameter extension;

* stable session resumption across topology or path changes.

TLS-DPA defines the handshake over an abstract TLS-DPA Channel. The channel only needs to provide:

* ordered or reliably framed delivery;

* a transport-level identifier (e.g., a TCP 4-tuple, QUIC Connection ID; [RFC9000] , or a UZP SessionID);

* uniqueness sufficient for transcript binding.

6. Transport-Agnostic Channel Model

TLS-DPA treats the underlying transport as providing one or more channels:

* *TCP* : traditional byte stream;

* *QUIC* : stream over a QUIC connection, identified by QUIC Connection ID ([RFC9000]);

* *UZP* : stream inside a ZPIT, identified by a UZP SessionID ([UZP]).

The handshake binds to this transport using the tlsdpa_transport_binding extension (see Section 10.2).

Figure 1: TLS-DPA operating over an abstract transport channel.

This figure places TLS-DPA above a transport channel to highlight the separation from the underlying relay.

7. Identity Binding Model

TLS-DPA authenticates peers using Service Identities, which may be:

* DNS names (validated per RFC 6125).
* UZP CIDs (canonical identities, derived from long-term public keys).
* UZP EIDs (ephemeral, session-level identities).
* UZPIF selectors resolved via Pantheon [UZPIF].

The Service Identity MUST be included in the handshake transcript and validated as described in Section Section 14.

CIDs are intended to be stable over meaningful operational time-scales: changes in CID MUST be treated as key-rotation events and not as transient transport artefacts.

8. Post-Quantum Key Exchange Model

TLS-DPA introduces unified hybrid-KEM negotiation via the tlsdpa_pq_kem_params extension. Supported KEM schemes include:

* X25519 (classical ECDH).
* Kyber768 (PQC KEM candidate).
* A hybrid X25519+Kyber768 mode.

The key schedule incorporates PQ KEM inputs prior to traffic secret derivation, following general design principles for hybrid KEMs in the NIST PQC process [NIST-PQC].

9. Key Schedule Summary

TLS-DPA modifies the TLS 1.3 key derivation [RFC8446] to include:

* Service Identity.
* Transport Binding.
* PQ KEM materials.

Exporter values MUST be bound to both identity and transport (Section Section 12).

At a high level, PQ hybrid KEM inputs augment the TLS 1.3 key schedule:

```
shared_secret = HKDF-Extract(kem_secret || ecdh_secret)
```

Figure 2: Hybrid shared secret extraction.

This equation shows the hybrid extraction step that combines KEM and ECDH inputs.

AEAD algorithms used with TLS-DPA MUST follow their specification-defined tag lengths. Tags MUST NOT be truncated below 96 bits, and 128-bit tags SHOULD be preferred where supported.

10. Extensions

(Conversion note) The extension names and structures in this document are intended for experimentation. This draft does not request IANA allocations. Where appropriate, implementations may use private-use ranges or negotiated profiles.

10.1. tlsdpa_service_identity

The tlsdpa_service_identity extension carries the Service Identity to which the TLS-DPA handshake is bound. For experimentation, this document uses an example private-use code point value (0xFE01); deployments MAY select alternative values by profile.

```
extension_type = 0xFE01

struct {
    ServiceIdentityType identity_type;
    opaque identity_value<1..2^16-1>;
} ServiceIdentity;

enum {
    dns_name(0),
    uzp_cid(1),
    uzp_eid(2),
    uzpif_selector(3),
    (255)
} ServiceIdentityType;
```

Figure 3: Service Identity extension structure (informative C-like syntax).

This figure shows the fields carried in the Service Identity extension.

The client MUST send exactly one Service Identity. The server MUST validate it according to its type (Section Section 14).

10.2. tlsdpa_transport_binding

The tlsdpa_transport_binding extension binds the handshake transcript to an underlying transport identifier and transport class. For experimentation, this document uses an example private-use code point value (0xFE02); deployments MAY select alternative values by profile.

```
extension_type = 0xFE02

struct {
    opaque transport_id<1..32>;
    uint8  transport_class;   /* 0=TCP, 1=QUIC, 2=UZP */
    opaque transport_params<0..256>;
} TransportBinding;
```

Figure 4: Transport Binding extension structure (informative C-like syntax).

This figure shows the fields used to bind the handshake to a transport identifier and class.

For UZP, transport_id MUST contain the UZP SessionID. For QUIC, it SHOULD contain the QUIC Connection ID [RFC9000].

10.3. tlsdpa_pq_kem_params

The tlsdpa_pq_kem_params extension carries the list of acceptable KEM schemes and related profile parameters. For experimentation, this document uses an example private-use code point value (0xFE03); deployments MAY select alternative values by profile.

```
extension_type = 0xFE03

struct {
    KEMScheme kem_list<2..2^8-1>;
} PQKemParams;

enum {
    x25519(0),
    kyber768(1),
    hybrid_x25519_kyber768(2),
    (255)
} KEMScheme;
```

Figure 5: PQ KEM parameters extension structure (informative C-like syntax).

This figure enumerates the acceptable KEM schemes and profile parameters.

The client proposes a list of acceptable KEM schemes. The selected scheme feeds into the key schedule.

11. Transcript Hashing Rules

New transcript components MUST be inserted as follows:

```
th = Hash(ClientHello
          || ServiceIdentity
          || TransportBinding
          || PQKemParams
          || ServerHello
          || ... )
```

Figure 6: Transcript hashing with identity and transport binding (illustrative).

This figure shows where the new identity and transport inputs are inserted into the transcript hash.

Hash mismatches MUST abort the handshake with illegal_transport_binding or identity_mismatch (Section Section 15).

12. Key Schedule and Exporters

PQ hybrid KEM inputs augment the TLS 1.3 key schedule as:

```
shared_secret = HKDF-Extract(kem_secret || ecdh_secret)
```

Figure 7: Hybrid shared secret extraction (illustrative).

This figure shows the shared secret input used for exporter derivation.

Exporter keys MUST incorporate identity and transport bindings.

12.1. Exporter Binding Requirements

TLS-DPA exporters MUST include:

* Service Identity (SID).

* Transport Binding (TB).

* PQ KEM scheme identifier.

* UZP SessionID (if transport_class = UZP).

12.2. Exporter Label Structure

```
label = "tlsdpa exporter" || 0x00 ||
  identity_type || transport_class
```

Figure 8: Exporter label structure (illustrative).

This figure shows the label composition that binds exporter output to identity and transport.

where identity_type is from ServiceIdentityType, and transport_class is from TransportBinding.

12.3. Exporter Context Structure

```
struct {
    opaque sid_hash[32];      /* BLAKE3-256 of Service Identity */
    opaque tb_hash[32];       /* BLAKE3-256 of TransportBinding */
    opaque kem_id[1];         /* selected KEM scheme */
} ExporterContext;
```

Figure 9: ExporterContext structure (informative C-like syntax).

This figure shows the exporter context fields derived from Service Identity, TransportBinding, and the selected KEM.

12.4. Example Exporter Computation

```
shared = HKDF-Extract(kem_secret || ecdh_secret);
ctx = ExporterContext(sid_hash, tb_hash, kem_id);
key = HKDF-Expand(shared, label, ctx, outlen);
```

Figure 10: Example exporter computation (illustrative).

This figure summarizes the exporter computation flow from shared secret to derived key.

13. Handshake Diagrams

13.1. Full UZP Flight Diagram

Figure Figure 11 provides an illustrative end-to-end view of a TLS-DPA handshake relayed via a rendezvous node (RN) in a UZP fabric. The RN forwards handshake flights without decrypting them. Binding ensures the RN cannot replay or modify flows undetected.

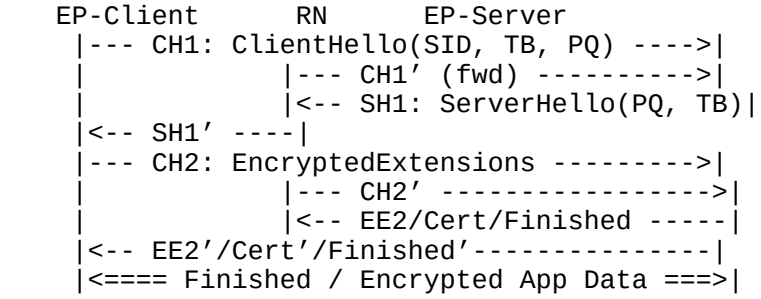

Figure 11: TLS-DPA handshake relayed via an RN, with end-to-end protection over the ZPIT (illustrative).

This figure traces the RN-relayed handshake flights while the endpoints retain end-to-end protection.

Where:

* SID: Service Identity.

* TB: Transport Binding (UZP SessionID mandatory).

* PQ: PQ KEM parameters.

13.2. Generalised TLS-DPA Flow

Figure Figure 12 shows a generalised view of the handshake and the role of a transport layer that relays flights but does not decrypt them.

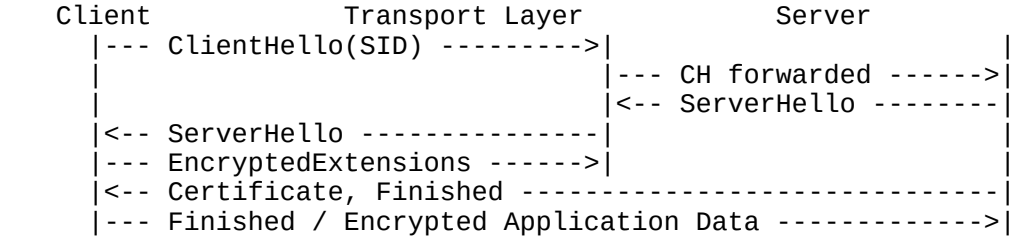

Figure 12: Generalised TLS-DPA handshake layers (illustrative).

This figure shows the transport relay separating TLS-DPA endpoints while preserving end-to-end security.

* SID is carried in the ClientHello.

* The transport layer relays flights but does not decrypt them.

* End-to-end Finished confirms key schedule integrity.

14. Service Identity Validation

14.1. DNS

DNS-based identities MUST be validated according to [RFC6125].

14.2. UZP CID

The CID MUST equal BLAKE3-256(server_longterm_public_key).

14.3. UZP EID

The EID MUST match the server-presented ephemeral identity for this session.

14.4. UZPIF Selector

UZPIF selectors MUST be resolved via Pantheon or local cached mappings consistent with Pantheon policy [UZPIF].

14.5. Failure Handling

If any validation fails, the implementation MUST abort the handshake with an appropriate alert (Section Section 15):

* identity_mismatch;

* illegal_transport_binding;

* pq_required.

15. New Alerts

TLS-DPA defines the following experimental alert descriptions for use in deployments and interoperability testing. The numeric values shown are illustrative and are not requested for IANA allocation by this draft.

```
enum {
    illegal_transport_binding(200),
    identity_mismatch(201),
    pq_required(202),
    grant_invalid(203),
    grant_expired(204),
    (255)
} AlertDescription;
```

Figure 13: Experimental alert descriptions (illustrative C-like syntax).

This figure lists the experimental alert codes defined by TLS-DPA.

16. Applicability to UZP / UZPIF

TLS-DPA maps naturally to UZP by binding:

* tlsdpa_service_identity -> UZP CID/EID.

* tlsdpa_transport_binding -> UZP SessionID.

* PQ capability and fallback -> Pantheon Grants.

UZP’s multi-step rendezvous and authentication model [UZP] and [UZPIF] provides:

* stronger pre-TLS identity establishment;

* reduced man-in-the-middle risk;

* deterministic channel binding for TLS-DPA.

17. Early Data (0-RTT)

Over UZP:

* early data is transmitted inside a ZPIT;

* replay protection uses CID/EID and Pantheon Grant metadata;

* early data MUST NOT be used if Pantheon Grants specify "no-replay".

17.1. RN Replay Detection

The RN MUST maintain a sliding replay cache keyed on:

```
GrantNonce || CID || EID || SessionID
```

Figure 14: Replay cache key tuple (illustrative).

This figure shows the tuple the RN uses to index replay state.

Entries MUST be retained for at least twice the maximum ZPIT propagation delay. Longer retention is permitted. If a duplicate early-flight tuple is observed, the RN MUST drop it silently.

17.2. Endpoint Replay Detection

Endpoints MUST track GrantNonce values associated with early data. For each:

```
(GrantNonce, CID, EID, ticket_age)
```

Figure 15: Endpoint replay tuple (illustrative).

This figure shows the endpoint tuple tracked to detect early data replay.

If an identical tuple is received twice within the resumption window, the endpoint MUST abort with illegal_parameter.

17.3. GrantNonce Interaction

Pantheon MUST issue a fresh GrantNonce per resumed or 0-RTT-enabled session. The nonce MUST be bound into the handshake transcript.

17.4. 0-RTT over UZP Rebinds

If the UZP SessionID changes during path migration, 0-RTT data MUST be rejected unless the new SessionID is verifiably linked to the previous one via Pantheon metadata.

18. Threat Model

TLS-DPA is designed to defend against:

* passive eavesdropping;
* active man-in-the-middle;
* downgrade attacks on both classical and PQ negotiation;
* identity spoofing;

* transport reattachment and rebinding attacks across overlays.

In UZP deployments, RN visibility is limited to flow identifiers and encrypted envelopes. No plaintext application data or Service Identity contents are exposed.

19. Operational Considerations

* Middleboxes SHOULD NOT assume fixed IP/port semantics for TLS-DPA channels.

* Monitoring SHOULD use exporter-based identity hooks rather than IP/port heuristics [NIST-SP800-207].

* Session resumption MUST accommodate overlay rebinds (e.g., QUIC Connection IDs, UZP SessionIDs).

* PQ keys and related metadata SHOULD be logged where required for compliance, in line with local policy.

20. Security Considerations

TLS-DPA implementations MUST:

1. ensure identity and transport bindings are transcript-authentic;

2. authenticate PQ hybrid negotiation and detect downgrades;

3. suppress downgrade unless explicitly permitted by policy;

4. minimise metadata exposure, especially in early flights;

5. prevent unauthorised reattachment across transports or overlays.

The threat model for TLS-DPA is discussed in Section Section 18.

21. IANA Considerations

This document does not request any IANA actions.

The example code points used for extension_type values and alert descriptions in this document are intended for experimentation (for example in private-use or locally coordinated deployments). Any future request for code point allocation is out of scope for this draft.

22. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <https://www.rfc-editor.org/info/rfc6125>.

23. Informative References

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <https://www.rfc-editor.org/info/rfc7401>.

[UZP] Fisher, B. A., "UZP: Universal Zero-Port Transport Protocol", Work in Progress, Internet-Draft, draft-dpa-uzp-transport, <https://datatracker.ietf.org/doc/html/draft-dpa-uzp-transport>.

[UZPIF] Fisher, B. A., "Universal Zero-Port Interconnect Framework (UZPIF)", Work in Progress, Internet-Draft, draft-dpa-uzpif-framework, <https://datatracker.ietf.org/doc/html/draft-dpa-uzpif-framework>.

[NIST-SP800-207] Rose, S., Borchert, O., Mitchell, S., and S. Connelly, "Zero Trust Architecture", NIST SP 800-207, 2019, <https://doi.org/10.6028/NIST.SP.800-207>.

[NIST-PQC] Technology, N. I. O. S. A., "NIST Post-Quantum Cryptography Standardization: Fourth Round Candidate Algorithms", 2022, <https://csrc.nist.gov/Projects/post-quantum-cryptography>.

Acknowledgements

The author thanks colleagues and early reviewers for discussions on identity-first security, transport binding, and post-quantum transition models. Any errors or omissions remain the author’s responsibility.

Author’s Address

Benjamin Anthony Fisher
DPA R&D Ltd (https://www.dpa-cloud.co.uk)
Email: b.fisher@dpa-cloud.co.uk
URI: https://orcid.org/0009-0004-4412-2269