Transport Layer Security
Internet-Draft
Updates: 8446 (if approved)
Intended status: Standards Track
Expires: 20 September 2026

D. Benjamin
Google LLC

19 March 2026

TLS Key Share Prediction
draft-ietf-tls-key-share-prediction-04

Abstract

This document defines a mechanism for servers to communicate supported key share algorithms in DNS. Clients may use this information to reduce TLS handshake round-trips.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://tlswg.github.io/tls-key-share-prediction/draft-ietf-tls-key-share-prediction.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-tls-key-share-prediction/.

Discussion of this document takes place on the Transport Layer Security Working Group mailing list (mailto:tls@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/tls/. Subscribe at https://www.ietf.org/mailman/listinfo/tls/.

Source for this draft and an issue tracker can be found at https://github.com/tlswg/tls-key-share-prediction.

Status of This Memo

This Internet-Draft will expire on 20 September 2026.

Table of Contents

1. Introduction

Named groups in TLS 1.3 [RFC8446] are negotiated with two lists in the ClientHello: The client sends its supported groups in the supported_groups extension, but also generates key shares for a subset in the key_share extension. Named groups in this subset can be used in one round trip, while named groups outside the subset require a HelloRetryRequest and hence two round trips. The additional round trip is undesirable for performance, but unused key shares consume network and computational resources, so clients often do not generate key shares for all groups.

Post-quantum key encapsulation methods (KEMs) have large keys and ciphertexts, so network costs are particularly pronounced. As a TLS ecosystem transitions from one post-quantum KEM to another, it is challenging to pick key shares without prior knowledge of the server's policies:

1. Predicting both post-quantum KEMs consumes excessive bandwidth on the unused option.

2. Predicting the old post-quantum KEM adds a round-trip cost to newer servers. Servers will be unlikely to transition as a result.

3. Predicting the new post-quantum KEM adds a round-trip cost to older servers. Particularly early in the transition, when most servers do not implement the new KEM, this may significantly regress performance.

This document defines a method for servers to declare their supported named groups in DNS, using SVCB or HTTPS resource records [RFC9460]. This allows the client to predict key shares more accurately.

2. Conventions and Definitions

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. DNS Service Parameter

This document defines the tls-supported-groups SvcParamKey [RFC9460], which specifies the endpoint’s supported TLS named groups, as a non-empty sequence of TLS NamedGroup codepoints in order of decreasing preference, with no duplicates. This allows clients connecting to the endpoint to reduce the likelihood of needing a HelloRetryRequest.

3.1. Format

The presentation value of the SvcParamValue is a non-empty comma-separated list (Appendix A.1 of [RFC9460]) of decimal integers between 0 and 65535 (inclusive) in ASCII, with no duplicate integers. Any other value is a syntax error. To enable simpler parsing, this SvcParam MUST NOT contain escape sequences.

The wire format of the SvcParamValue is a sequence of 2-octet numeric values in network byte order. An empty list of values is invalid, as is a list containing duplicates.

For example, a TLS server which prefers x25519 (29) and also supports secp256r1 (23) would add a tls-supported-groups SvcParamValue containing 29 and 23. The presentation value would be "29,23". The wire format of the SvcParamValue would be four octets, represented in hexadecimal as 001d0017.

The following is an example of the value appearing in a complete DNS record in the presentation syntax:

```
example.net.  7200  IN SVCB 3 server.example.net. (
    port="8004" tls-supported-groups=29,23 )
```

3.2. Configuring Services

Services SHOULD include supported TLS named groups, in order of decreasing preference in the tls-supported-groups parameter of their HTTPS or SVCB endpoints. As TLS configuration is updated, services SHOULD update the DNS record to match. Services MAY include GREASE values [RFC8701] in this list.

3.3. Client Behavior

When connecting to a service endpoint whose HTTPS or SVCB record contains the tls-supported-groups parameter, the client evaluates the server’s list against its configuration to predict which named group will be chosen. When evaluating the server’s list, the client MUST ignore any codepoints that it does not support or recognize.

If the client predicts a named group, the client SHOULD send a key_share extension containing just that named group in the initial ClientHello. The client MAY continue to send other key shares to reduce mispredictions (see Section 3.4), though this comes at additional network and computational cost. The client MAY also ignore the prediction, e.g., it chooses not to apply this process to some groups (see Section 4).

If there were no named groups in common, the client SHOULD proceed as if the tls-supported-groups parameter was not present and predict some default set of key shares. The HTTPS or SVCB record may have been stale, so it is possible the server still has a named group in common.

This process does not modify the supported_groups extension. To avoid downgrade attacks, the client MUST continue to send all its supported groups, in preference order, in supported_groups. See Section 4 for additional discussion on downgrades.

3.4. Misprediction

Although this service parameter is intended to reduce key share mispredictions, mispredictions may still occur in some scenarios. For example:

* The client has fetched a stale HTTPS or SVCB record that no longer reflects the server configuration

* The server is in the process of deploying a change to named group configuration, and different server instances temporarily evaluate different configuration

* The client was unable to fetch the HTTPS or SVCB record

* The client and server implement incompatible selection algorithms, such that client's evaluation of the service parameter did not match the server's final selection

Clients and servers MUST correctly handle mispredictions by responding to and sending HelloRetryRequest, respectively.

4. Security Considerations

This document introduces a mechanism for clients to vary the key_share extension based on DNS. DNS responses are unauthenticated in many deployments. An attacker may be able to forge an HTTPS or SVCB record and influence the client's predicted named groups. That, in turn, can influence the named group selected by the TLS server, as TLS's downgrade protections only extend to the ClientHello itself.

Provided the client's supported_groups list always reflects the unmodified client preference list, this is safe. The scope of attacker influence depends on how the server selects a group. Servers are expected to evaluate the combination of key_share and supported_groups according to their selection goals and the definitions in [RFC8446]. When deciding between multiple common groups, a server might consider:

* The server's local preferences, picking one it considers best.

* The client's preference order in supported_groups, picking one the client considers best.

* Which groups appear in key_share, picking one that avoids a HelloRetryRequest.

The last case, presence in key_share, is under attacker influence in this mechanism. However, Section 4.2.8 of [RFC8446] already permits the client to omit its most preferred groups in key_share. Servers are thus expected to only select by key_share when they opt to consider neither the client's preference nor their own. That is, it is only appropriate in cases where the two groups have comparable preference, such that round-trip costs dominate. Servers SHOULD NOT use key_share to select a classical named group over a post-quantum named group.

To reduce the risk of downgrade attacks with incorrectly deployed servers, clients MAY choose to ignore tls-supported-groups when the result would predict a less preferred group. For example, a client might prefer post-quantum groups, but support ECDH groups with older servers. It MAY then ignore DNS-based ECDH predictions, limiting tls-supported-groups to post-quantum options. In this case, transitions between post-quantum groups, where the bandwidth concerns are more pronounced, remain optimized, but ECDH-only servers cannot take advantage of tls-supported-groups.

5. IANA Considerations

This document updates the Service Parameter Keys registry [RFC9460] with the following entry:

Number	Name	Meaning	Format Reference	Change Controller
9	tls-supported-groups	Supported groups in TLS	(this document) Section 3.1	IETF

Table 1

6. 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/rfc/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/rfc/rfc8174>.

[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/rfc/rfc8446>.

[RFC8701] Benjamin, D., "Applying Generate Random Extensions And Sustain Extensibility (GREASE) to TLS Extensibility", RFC 8701, DOI 10.17487/RFC8701, January 2020, <https://www.rfc-editor.org/rfc/rfc8701>.

[RFC9460] Schwartz, B., Bishop, M., and E. Nygren, "Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)", RFC 9460, DOI 10.17487/RFC9460, November 2023, <https://www.rfc-editor.org/rfc/rfc9460>.

Acknowledgments

The author would like to thank David Adrian, Bob Beck, Marc Penninga, Sophie Schmieg, Martin Thomson, and Bas Westerbaan for discussions and review of this document.

Author’s Address

David Benjamin
Google LLC
Email: davidben@google.com