

Network Working Group  
Internet-Draft  
Updates: 9001 (if approved)  
Intended status: Standards Track  
Expires: 27 August 2025

Y. Rosomakho  
Zscaler  
H. Tschofenig  
H-BRS  
23 February 2025

Extended Key Update for QUIC  
draft-rosomakho-quic-extended-key-update-00

## Abstract

This document specifies an Extended Key Update mechanism for the QUIC protocol, building on the foundation of the TLS Extended Key Update. The TLS Extended Key Update specification enhances the TLS protocol by introducing key updates with forward secrecy, eliminating the need to perform a full handshake. This feature is particularly beneficial for maintaining security in scenarios involving long-lived connections.

This specification replaces the QUIC Key Update mechanism described in the "Using TLS to Secure QUIC" specification.

## 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://yaroslavros.github.io/quic-extended-key-update/draft-rosomakho-quic-extended-key-update.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-rosomakho-quic-extended-key-update/>.

Source for this draft and an issue tracker can be found at <https://github.com/yaroslavros/quic-extended-key-update>.

## 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 27 August 2025.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions and Definitions . . . . .	3
3. Extended Key Update Negotiation . . . . .	3
4. Extended Key Update Messages . . . . .	4
5. Updating the Traffic Secrets . . . . .	4
6. Security Considerations . . . . .	6
7. IANA Considerations . . . . .	6
8. References . . . . .	6
8.1. Normative References . . . . .	6
8.2. Informative References . . . . .	7
Acknowledgments . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

The QUIC protocol [QUIC] provides a secure, versatile transport for various applications, suitable for long-lived sessions in environments like industrial IoT, telecommunication networks or Virtual Private Networks (VPN), as specified in [RFC9484].

The TLS Extended Key Update [I-D.ietf-tls-extended-key-update] introduces a mechanism to enhance the security and flexibility of encrypted communication protocols by enabling frequent key updates without requiring a full handshake renegotiation. This approach allows applications to refresh their encryption keys more often,

improving forward secrecy and reducing the risk of key compromise over long-lived connections. By separating key updates from the computationally expensive handshake process, the specification provides a lightweight method for maintaining robust encryption in scenarios where connections need to remain secure for extended periods.

This new TLS capability is particularly valuable in environments where interruptions to perform a full key exchange would cause significant disruption. Other encrypted communication protocols, such as IPsec [IKEv2] and SSH [SSH-TRANSPORT], include mechanisms for re-exchanging keys without interrupting active sessions. The TLS Extended Key Update specification ensures that even in the face of potential key compromise, sensitive data remains protected due to frequent key rotation and the use of forward secrecy.

This specification builds on concepts from [I-D.ietf-tls-extended-key-update] and applies them to the QUIC protocol context. It thereby replaces the QUIC Key Update mechanism described in Section 6 of [QUIC-TLS].

## 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.

Readers are assumed to be familiar with [I-D.ietf-tls-extended-key-update].

## 3. Extended Key Update Negotiation

QUIC peers negotiate Extended Key Update through the TLS handshake process, as outlined in Section 4 of [I-D.ietf-tls-extended-key-update]. Extended Key Update MUST NOT be used unless both QUIC peers include the TLS flags extension [I-D.ietf-tls-tlsflags] in the handshake and set the "Extended\_Key\_Update" flag.

Once Extended Key Update has been successfully negotiated, QUIC peers MUST only use the Extended Key Update process defined in this document. The standard QUIC Key Update mechanism from Section 6 of [QUIC-TLS] MUST NOT be used for the duration of the session, as both Key Update and Extended Key Update use the Key Phase bit to signal the use of updated keys. The Key Phase bit is initially set to 0 and toggled to indicate a key update following the successful post-handshake exchange of Extended Key Update messages.

#### 4. Extended Key Update Messages

Either party MAY initiate the Extended Key Update process by sending an ExtendedKeyUpdateRequest TLS handshake message in a QUIC CRYPTO frame. This message MUST NOT be sent before the QUIC handshake is confirmed, as described in Section 4.1.2 of [QUIC-TLS], or before previous Extended Key Update process is complete. If a QUIC endpoint receives an ExtendedKeyUpdateRequest message before the QUIC handshake is complete or during ongoing prior Extended Key Update, it MUST terminate the connection with an error of type 0x010a, equivalent to TLS unexpected\_message alert as specified in Section 4.8 of [QUIC-TLS].

Upon receiving an ExtendedKeyUpdateRequest, the recipient MUST respond with an ExtendedKeyUpdateResponse TLS handshake message within a QUIC CRYPTO frame. If a QUIC endpoint receives an unexpected ExtendedKeyUpdateResponse message from its peer, it MUST treat this as a TLS protocol error and terminate the connection.

Specifications of the ExtendedKeyUpdateRequest and ExtendedKeyUpdateResponse messages are provided in Section 5 of [I-D.ietf-tls-extended-key-update]. Any mismatch between the negotiated NamedGroup during the initial handshake and the group used in the Extended Key Update message, or an incorrect length of the encapsulated key MUST result in connection termination with error of type 0x012f, equivalent to TLS illegal\_parameter alert.

If the Extended Key Update initiator receives a retry or rejected status in the ExtendedKeyUpdateResponse message, it MAY terminate the connection with error of type 0x01TBD, equivalent to TLS extended\_key\_update\_required alert.

#### 5. Updating the Traffic Secrets

After sending an ExtendedKeyUpdateResponse with accepted status, the responder derives new packet protection traffic secrets. The responder MUST continue using the previous secrets until it has received a packet with the Key Phase bit flipped and has successfully unprotected it using the new keys.

After receiving and successfully processing an `ExtendedKeyUpdateResponse` with accepted status, the initiator derives new packet protection traffic secrets, flips the Key Phase bit for new packets, and uses the new write secret to protect them. The initiator **MUST** retain the old read secret until it has received a packet with a flipped Key Phase bit from the responder and successfully unprotected it using the new read secret.

Both endpoints **SHOULD** retain old read secrets for some time after unprotecting a packet encrypted with the new keys. Discarding old secret too early may cause delayed packets to be discarded, which the peer may interpret as packet loss, potentially impacting performance.

Figure 1 shows this interaction graphically.

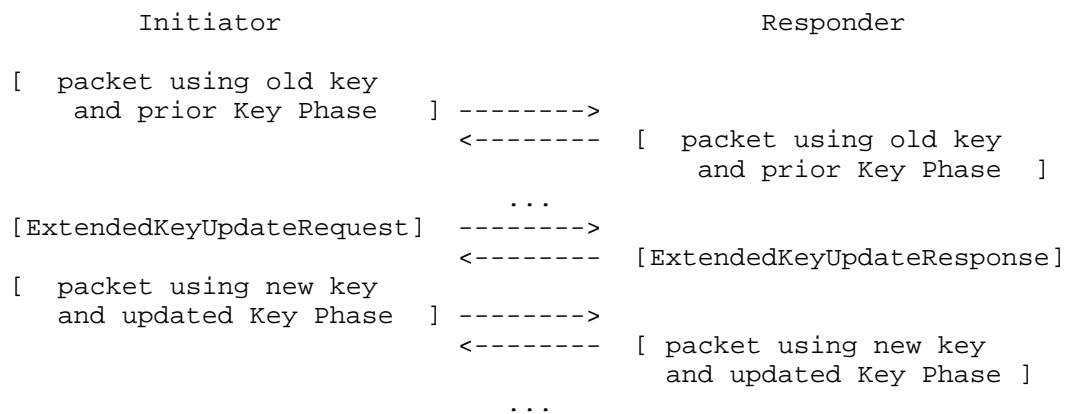


Figure 1: Extended Key Update Process in QUIC.

QUIC endpoints **MUST NOT** send `NewKeyUpdate` TLS handshake messages, defined in [I-D.ietf-tls-extended-key-update], and instead relies on the use of the Key Phase bit. Endpoints **MUST** treat the receipt of a TLS `NewKeyUpdate` message as a connection error of type 0x010a. QUIC endpoints that have agreed to the Extended Key Update process **MUST NOT** change the Key Phase bit without a successful exchange of Extended Key Update TLS messages. Receiving a packet with the Key Phase bit changed without a successful Extended Key Update exchange **MUST** be treated as a connection error of type `KEY_UPDATE_ERROR` (0x0e).

The design of the key derivation function for computing the next generation of secrets corresponds to the one described in Section 6 of [I-D.ietf-tls-extended-key-update] with the exception of the use of a different label.

```
sk = HKDF-Extract(Transcript-Hash(ExtendedKeyUpdateRequest,  
                                ExtendedKeyUpdateResponse), secret)  
  
secret_<n+1> = HKDF-Expand-Label(sk, "quic eku",  
                                secret_<n>, Hash.length)
```

The corresponding key and IV are derived from the new secret as defined in Section 5.1 of [QUIC-TLS]. The header protection key is not updated.

## 6. Security Considerations

This specification describes an update to the key schedule of QUIC. Therefore, implementations MUST ensure that peers adhere strictly to the process described in this document. Packets with higher packet numbers MUST NOT be protected using an older generation of secrets, as this could compromise key synchronization and security.

As key exchange may be computationally intensive, responders SHOULD consider rate-limiting Extended Key Exchange requests. This can be done by responding with retry status as outlined in Section 5 of [I-D.ietf-tls-extended-key-update] and terminating connections for initiators that violate the back-off timer. This approach helps prevent excessive load on endpoints and mitigates the risk of denial-of-service attacks.

## 7. IANA Considerations

This document has no IANA actions.

## 8. References

### 8.1. Normative References

[I-D.ietf-tls-extended-key-update]  
Tschofenig, H., T端xen, M., Reddy.K, T., Fries, S., and Y. Rosomakho, "Extended Key Update for Transport Layer Security (TLS) 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-extended-key-update-03, 21 October 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-tls-extended-key-update-03>>.

[I-D.ietf-tls-tlsflags]  
Nir, Y., "A Flags Extension for TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-tlsflags-14, 13 September 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-tls-tlsflags-14>>.

- [QUIC] 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/rfc/rfc9000>>.
- [QUIC-TLS] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <<https://www.rfc-editor.org/rfc/rfc9001>>.
- [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>>.

## 8.2. Informative References

- [IKEv2] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<https://www.rfc-editor.org/rfc/rfc7296>>.
- [RFC9484] Pauly, T., Ed., Schinazi, D., Chernyakhovsky, A., K端hlewind, M., and M. Westerlund, "Proxying IP in HTTP", RFC 9484, DOI 10.17487/RFC9484, October 2023, <<https://www.rfc-editor.org/rfc/rfc9484>>.
- [SSH-TRANSPORT] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <<https://www.rfc-editor.org/rfc/rfc4253>>.

## Acknowledgments

We would like to thank Martin Thomson and Sean Turner for their early review of this design and their invaluable feedback.

## Authors' Addresses

Yaroslav Rosomakho  
Zscaler  
Email: [yrosomakho@zscaler.com](mailto:yrosomakho@zscaler.com)

Hannes Tschofenig  
University of Applied Sciences Bonn-Rhein-Sieg  
Germany  
Email: Hannes.Tschofenig@gmx.net