TCPM
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

M. Baerts
UCLouvain
O. Bonaventure
UCLouvain & WELRI

7 July 2025

Multipath TCP with external keys
draft-baerts-tcpm-mptcpext-00

Abstract

This document proposes an extension to Multipath TCP that allows application layer protocols such as TLS or SSH to provide keys to authenticate the creation of new subflows.

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://ipnetworkinglab.github.io/draft-mptcp-ext/draft-baerts-tcpm-mptcpext.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-baerts-tcpm-mptcpext/.

Discussion of this document takes place on the TCPM Individual mailing list (mailto:tcpm@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/tcpm/. Subscribe at https://www.ietf.org/mailman/listinfo/tcpm/.

Source for this draft and an issue tracker can be found at https://github.com/IPNetworkingLab/draft-mptcp-ext.

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 8 January 2026.

Table of Contents

1. Introduction

This document addresses an important limitation of Multipath TCP [RFC8684]: the exchange of plain text keys during the handshake.

From a security viewpoint, Multipath TCP is vulnerable to on-path attacks [RFC6181]. Since Multipath TCP relies on keys that are exchanged in clear during the handshake, an on-path attacker can easily collect the authentication keys and later establish a subflow on an existing Multipath TCP connection. If this connection is used to support secure protocols such as TLS [RFC8446] or SSH [RFC4253], the attacker will only be able to disrupt the connection.

This document proposes a modification to the MP_CAPABLE and MP_JOIN options that enables Multipath TCP hosts to use keys that are derived by upper layer protocols such as TLS or SSH. This idea has already been discussed in the past [I-D.paasch-mptcp-ssl-00], [I-D.paasch-mptcp-application-authentication-00] and [I-D.paasch-mptcp-tls-authentication-00]. We provide an overview of this extension in Section 3 and describe the protocol modifications in Section 4.

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. Protocol overview

This document proposes an extension that allows Multipath TCP to use either a pre-configured key or a key derived by an upper layer security protocol to authenticate the advertisement of additional addresses, the establishment of new subflows, and the abrupt closure of a whole connection. This extension is negotiated during the establishment of the Multipath TCP connection by setting the TBD bit in the MP_CAPABLE option. This is illustrated in Figure 1.

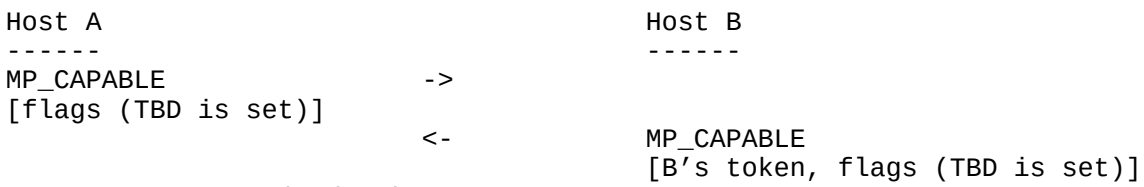

Figure 1: Negotiation of the utilization of external keys

If the TBD flag is set and the responder supports the option, it returns an MP_CAPABLE that contains the 32-bit token that it uses to identify this connection. The connection initiator replies with an MP_CAPABLE option containing its 32-bit token and the remote 32-bit token.

This modification has two important advantages compared to Multipath TCP version 1 [RFC8684]. First, the MP_CAPABLE option is shorter. It contains only two 32-bit tokens instead of two 64-bit keys in the third ACK. Second, the token is not derived from a random key using

a hash function. This implies that there is no risk of collision between a new key and a token used for an existing connection. The token must uniquely identify the associated connection and should be selected randomly [RFC4086].

After the handshake, host A and host B cannot create additional subflows or exchange additional addresses. These operations can only occur once they have agreed on an external key. Once a host has learned an external key (e.g. through configuration, socket option, or derived from a security protocol), it SHOULD inform the other by sending a hash of this key in a NEW_KEY option a shown in Figure 2. The key is installed once a host has received a hash of the key from the other host.

Security protocols need to change keys regularly for security reasons. Multipath TCP version 1 [RFC8684] does not support changing the security keys. This extension uses a key identifier to support key changes. All authenticated options contain the K bit which is the identifier of the key used to authenticate it. The initial external key corresponds to identifier 0.

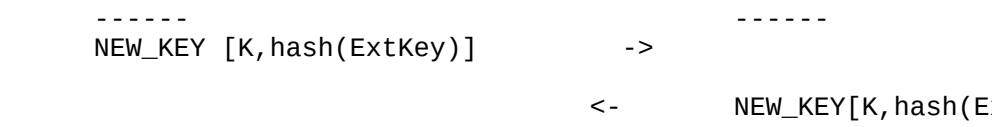

Figure 2: Confirmation of the utilization of a new external key

The key identifier is included in the modified ADD_ADDR, MP_JOIN and FASTCLOSE options described later in this document.

4. Changes to Multipath TCP

This section describes the changes to the Multipath TCP options that are required to support external keys.

4.1. Extending the MP_CAPABLE and MP_JOIN options

[RFC8684] defines the MP_CAPABLE option as shown in Figure 3.

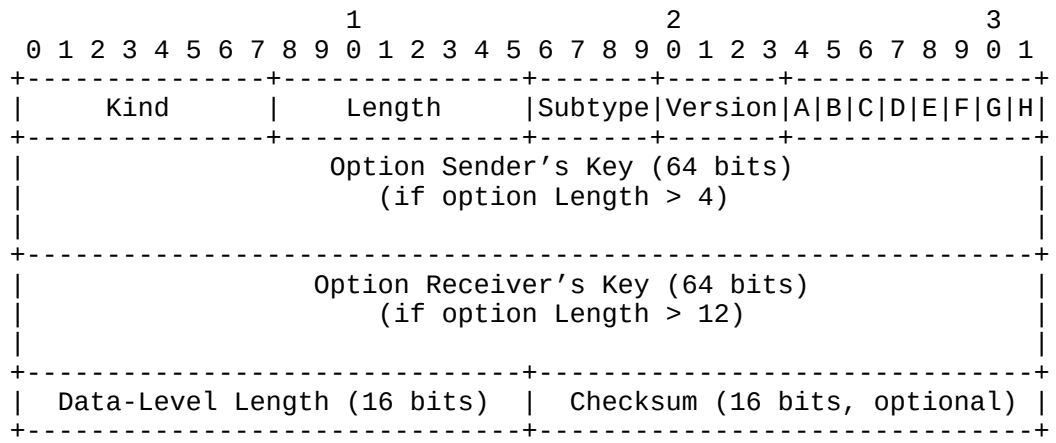

Figure 3: The MP_CAPABLE option in RFC8684

This option contains several flags, A-H. Flags A, B, C, and H are specified in [RFC8684]. This document changes the TBD Flag. If this Flag is set in a SYN, this indicates the utilization of external keys. An external key is a key which is either known, e.g. by configuration as a shared secret, or derived from a negotiated secure key, e.g. by protocols such as SSH or TLS. This key is used as an authentication key for the establishment of additional subflows.

A Multipath TCP implementation maintains two 64-bit keys:

* a local key chosen by the host and exchanged during the handshake

* a remote key learned during the handshake

As specified in [RFC8684], a local 32-bit token and a remote 32-bit token are derived from these keys. The keys and the token are known at the end of the handshake.

When the external keys are used, the situation is different. The connection initiator sends an empty MP_CAPABLE option in its SYN segment. A responder that receives a SYN with the MP_CAPABLE option having the TBD bit set responds with an MP_CAPABLE option and the TBD bit set if it supports the external keys. Otherwise, it replies with an MP_CAPABLE option whose TBD bit is reset and follows the procedure defined in [RFC8684].

If the responder replies with an MP_CAPABLE option whose TBD Flag is set, the option in the SYN+ACK contains the 32-bit token that it uses to identify this connection.

Upon reception of the SYN+ACK, the connection initiator replies with a third ACK that contains an MP_CAPABLE option with the TBD bit set. This option contains the initiator and the responder tokens.

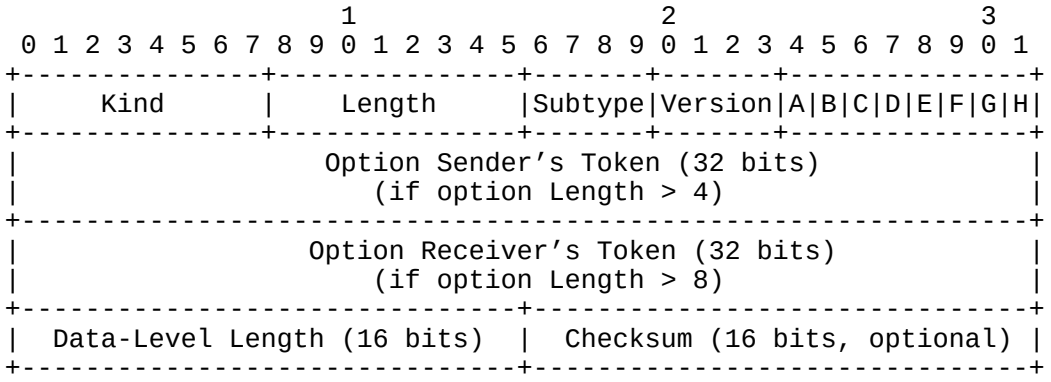

Figure 4: The modified MP_CAPABLE option

If the TBD bit was set in the MP_CAPABLE option of the SYN+ACK, this indicates that there are no security keys associated with the connection. This implies that it is impossible to advertise addresses or join an additional subflow until external keys have been exchanged.

4.2. Changing the external key

Once the Multipath TCP connection has been established, the applications can decide to use an external key.

Once the hosts have agreed on an external key to use to authenticate the MP_JOIN, ADD_ADDR, and MP_FASTCLOSE options on a connection, they inform the other host by sending a NEW_KEY option. This option is shown in Figure 5.

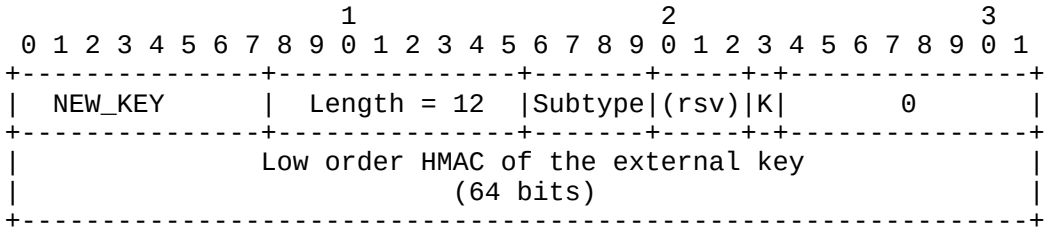

Figure 5: The NEW_KEY option

This new option contains two pieces of information:

* a one-bit key identifier (K)

* a 64-bit HMAC of the external key

The key identifier identifies the last key exchanged on a connection. The key identifiers start at 0, i.e. the first NEW_KEY option contains the K bit set to zero. The 64-bit HMAC contains the low order 64 bits of a HMAC of the external key computed using the negotiated hash algorithm. The key for this HMAC, in the case of a message transmitted by Host A, is Token-A followed by Token-B; and in the case of Host B, Key-B followed by Key-A. The "message" for the HMAC algorithm in each case is the external key.

Once a host has sent a NEW_KEY option, it SHOULD start a timer. If it does not receive an option containing the same hash value, it should retransmit the option.

A host must store two external keys:

* the current one

* the next key

A key is considered to be active once a host has received a NEW_KEY option containing a HMAC of this key. If a host receives a NEW_KEY option whose HMAC and key identifier do not match the stored ones, it simply discards the option.

4.3. Using the external key

While Multipath TCP version 1 uses two different keys announced by the communicating hosts, the external key is a key shared by both hosts. [RFC8684] defined several procedures that rely on these two keys to authenticate the establishment of subflows using the MP_JOIN option, the advertisement for new addresses, or the fast termination of a connection.

These procedures change with an external key. The first modification is that these options now contain a K bit that indicates the identifier of the external key used to (request to) authenticate the option. The second modification is that instead of computing HMACs over KeyA||KeyB, the HMACs defined in [RFC8684] are now computed using the external key whose identifier is K.

The new format of the MP_JOIN option is shown in Figure 6.

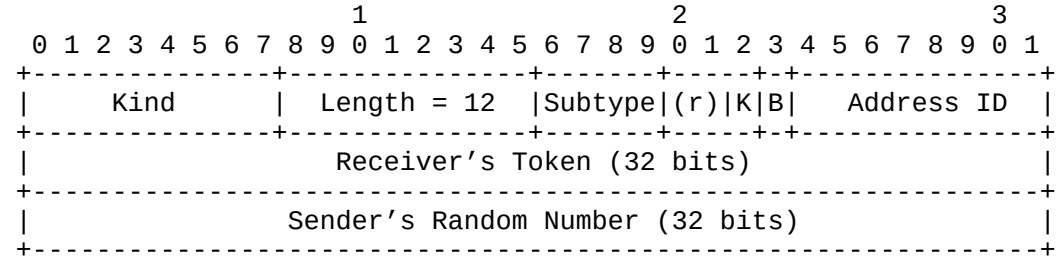

Figure 6: The modified MP_JOIN option

The new format of the ADD_ADDR option is shown in Figure 7.

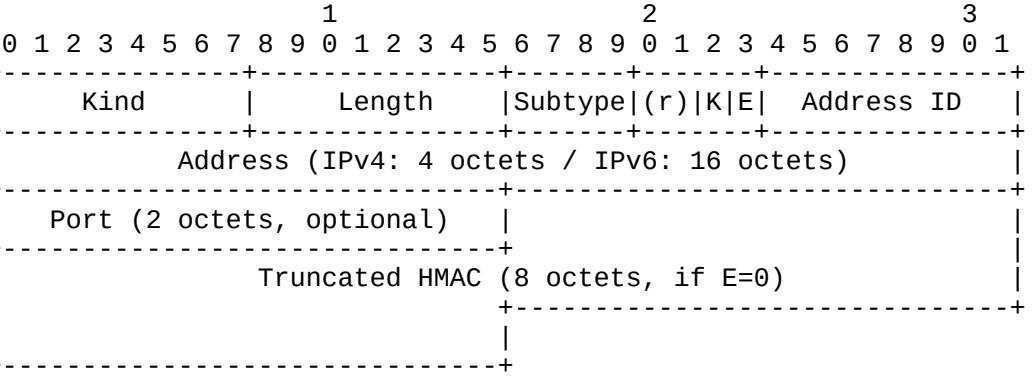

Figure 7: The modified ADD_ADDR option

The modified format of the MP_FASTCLOSE option is shown in Figure 8. This option does no longer contain the receivers' key as in [RFC8684]. Instead, it contains a truncated HMAC of the external key. The key for this HMAC is, in the case of a message transmitted by Host A, Token-A followed by Token-B; and in the case of Host B, Token-B followed by Token-A. The "message" for the HMAC algorithm is, in each case, the external key. The K bit indicates the corresponding key identifier.

A host can still reset an MPTCP connection before the initial external keys got exchanged, while there is only one subflow then. This SHOULD be done by sending a TCP RST.

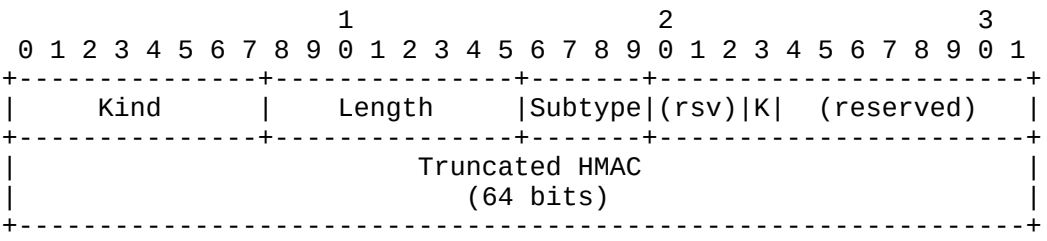

Figure 8: The modified MP_FASTCLOSE option

5. Security Considerations

The solution proposed in this document aims at preventing the attacks where an on-path attacker observes the keys associated with a Multipath TCP connection. Since these keys are not exposed anymore, attackers cannot use them to add subflows to an existing Multipath TCP connection.

6. IANA Considerations

This document requests the IANA to reserve flag TBD of the MP_CAPABLE option as defined in this document. It proposes to use the E flag. It also proposes to add the K bit to the MP_JOIN, ADD_ADDR, and MP_FASTCLOSE options. Finally, it defines the NEW_KEY option. Subtype 0x9 is suggested for this option.

7. References

7.1. 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>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/rfc/rfc4086>.

[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>.

[RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March 2020, <https://www.rfc-editor.org/rfc/rfc8684>.

7.2. Informative References

[I-D.paasch-mptcp-application-authentication-00]
Paasch, C. and A. Ford, "Application Layer Authentication for MPTCP", Work in Progress, Internet-Draft, draft-paasch-mptcp-application-authentication-00, 27 May 2016, <https://datatracker.ietf.org/doc/html/draft-paasch-mptcp-application-authentication-00>.

[I-D.paasch-mptcp-ssl-00]
Paasch, C. and O. Bonaventure, "Securing the MultiPath TCP handshake with external keys", Work in Progress, Internet-Draft, draft-paasch-mptcp-ssl-00, 15 October 2012, <https://datatracker.ietf.org/doc/html/draft-paasch-mptcp-ssl-00>.

[I-D.paasch-mptcp-tls-authentication-00]
Paasch, C. and A. Ford, "TLS Authentication for MPTCP", Work in Progress, Internet-Draft, draft-paasch-mptcp-tls-authentication-00, 27 May 2016, <https://datatracker.ietf.org/doc/html/draft-paasch-mptcp-tls-authentication-00>.

[RFC4253] 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>.

[RFC6181] Bagnulo, M., "Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6181, DOI 10.17487/RFC6181, March 2011, <https://www.rfc-editor.org/rfc/rfc6181>.

[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>.

Acknowledgments

This work was supported by the Walloon government within the FRFS-WEL-T SEEIP project. The idea of using external keys to secure Multipath TCP was initially proposed in [I-D.paasch-mptcp-ssl-00].

Authors' Addresses

Matthieu Baerts
UCLouvain
Email: matthieu.baerts@uclouvain.be

Olivier Bonaventure
UCLouvain & WELRI
Email: olivier.bonaventure@uclouvain.be