

TLS
Fossati
Internet-Draft
Ed.
Updates: [6347](#) (if approved)
Limited
Intended status: Standards Track
2020
Expires: September 3, 2020

T.
H. Tschofenig,
Arm
March 2,

**Return Routability Check for DTLS 1.2 and DTLS 1.3
draft-tschofenig-tls-dtls-rrc-01**

Abstract

This document specifies a return routability check for use in context of the Connection ID (CID) construct for the Datagram Transport Layer Security (DTLS) protocol versions 1.2 and 1.3.

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 September 3, 2020.

Copyright Notice

Copyright (c) 2020 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

- [1](#) 1. Introduction
- [2](#) 2. Conventions and Terminology
- [3](#) 3. The Return Routability Check Message
- [3](#) 4. RRC Example
- [4](#) 5. Security and Privacy Considerations
- [7](#) 6. IANA Considerations
- [7](#) 7. Open Issues
- [7](#) 8. Normative References
- [7](#) [Appendix A](#). History
- [8](#) [Appendix B](#). Acknowledgements
- [8](#) Authors' Addresses
- [8](#)

1. Introduction

In "classical" DTLS, selecting a security context of an incoming DTLS record is accomplished with the help of the 5-tuple, i.e. source IP address, source port, transport protocol, destination IP address, and destination port. Changes to this 5 tuple can happen for a variety reasons over the lifetime of the DTLS session. In the IoT context, NAT rebinding is common with sleepy devices. Other examples include end host mobility and multi-homing. Without CID, if the source IP address and/or source port changes during the lifetime of an ongoing DTLS session then the receiver will be unable to locate the correct security context. As a result, the DTLS handshake has to be re-run. Of course, it is not necessary to re-run the full handshake if

session resumption is supported and negotiated.

A CID is an identifier carried in the record layer header of a DTLS datagram that gives the receiver additional information for selecting the appropriate security context. The CID mechanism has been specified in [[I-D.ietf-tls-dtls-connection-id](#)] for DTLS 1.2 and in [[I-D.ietf-tls-dtls13](#)] for DTLS 1.3.

Section 6 of [[I-D.ietf-tls-dtls-connection-id](#)] describes how the use of CID increases the attack surface by providing both on-path and off-path attackers an opportunity for (D)DoS. It then goes on describing the steps a DTLS principal must take when a record with a CID is received that has a source address (and/or port) different from the one currently associated with the DTLS connection.

However,

the actual mechanism for ensuring that the new peer address is willing to receive and process DTLS records is left open. This document standardizes a return routability check (RRC) as part of

the

DTLS protocol itself.

The return routability check is performed by the receiving peer before the CID-to-IP address/port binding is updated in that peer's session state database. This is done in order to provide more confidence to the receiving peer that the sending peer is reachable at the indicated address and port.

2. 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

[BCP](#)

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

This document assumes familiarity with the CID format and protocol defined for DTLS 1.2 [[I-D.ietf-tls-dtls-connection-id](#)] and for DTLS 1.3 [[I-D.ietf-tls-dtls13](#)].

3. The Return Routability Check Message

When a record with CID is received that has the source address of the

enclosing UDP datagram different from the one previously associated with that CID, the receiver MUST NOT update its view of the peer's

IP

address and port number with the source specified in the UDP

datagram

before cryptographically validating the enclosed record(s) but instead perform a return routability check.

```
enum {
    invalid(0),
    change_cipher_spec(20),
    alert(21),
    handshake(22),
    application_data(23),
    heartbeat(24), /* RFC 6520 */
    return_routability_check(TBD), /* NEW */
}
```

(255)
} ContentType;

Fossati & Tschofenig Expires September 3, 2020
3]

[Page

```
struct {  
    opaque cookie<1..2^16-1>;  
} Cookie;  
  
struct {  
    Cookie cookie;  
} return_routability_check;
```

The newly introduced `return_routability_check` message contains a cookie. The semantic of the cookie is similar to the cookie used in the `HelloRetryRequest` message defined in [[RFC8446](#)].

The `return_routability_check` message MUST be authenticated and encrypted using the currently active security context.

The receiver that observes the peer's address and or port update MUST

stop sending any buffered application data (or limit the sending rate to a TBD threshold) and initiate the return routability check that proceeds as follows:

1. A cookie is placed in the `return_routability_check` message;
2. The message is sent to the observed new address and a timeout `T` is started;
3. The peer endpoint, after successfully verifying the received `return_routability_check` message echoes it back;
4. When the initiator receives and verifies the `return_routability_check` message, it updates the peer address binding;
5. If `T` expires, or the address confirmation fails, the peer address binding is not updated.

After this point, any pending send operation is resumed to the bound peer address.

4. RRC Example

The example shown in Figure 1 illustrates a client and a server exchanging application payloads protected by DTLS with an unilaterally used CIDs. At some point in the communication interaction the IP address used by the client changes and, thanks to the CID usage, the security context to interpret the record is successfully located by the server. However, the server wants to test the reachability of the client at his new IP address, to avoid

being abused (e.g., as an amplifier) by an attacker impersonating the client.

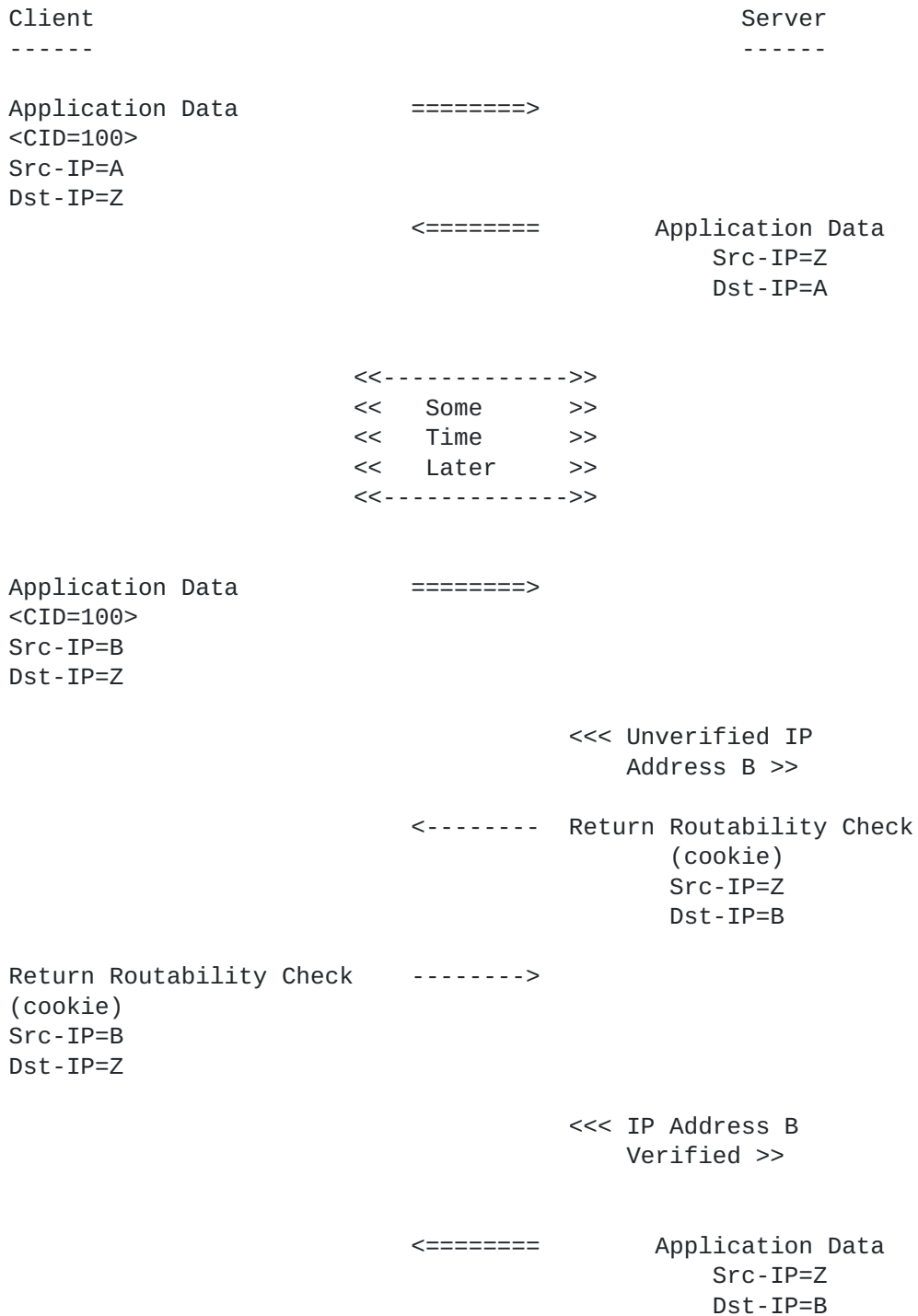


Figure 1: Return Routability Example

5. Security and Privacy Considerations

Note that the return routability checks do not protect against flooding of third-parties if the attacker is on-path, as the attacker

can redirect the return routability checks to the real peer (even if those datagrams are cryptographically authenticated). On-path adversaries can, in general, pose a harm to connectivity.

6. IANA Considerations

IANA is requested to allocate an entry to the existing TLS "ContentType" registry, for the return_routability_check(TBD) defined in this document.

7. Open Issues

- Should the return routability check use separate sequence numbers and replay windows?
- Should the heartbeat message be re-used instead of the proposed new message exchange?

8. Normative References

[I-D.ietf-tls-dtls-connection-id]

Rescorla, E., Tschofenig, H., and T. Fossati, "Connection Identifiers for DTLS 1.2", [draft-ietf-tls-dtls-](#)

[connection-](#)

[id-07](#) (work in progress), October 2019.

[I-D.ietf-tls-dtls13]

Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", [draft-ietf-tls-dtls13-34](#) (work in progress), November 2019.

[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>>.

[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>>.

Fossati & Tschofenig Expires September 3, 2020
7]

[Page

Appendix A. History

RFC EDITOR: PLEASE REMOVE THE THIS SECTION

- 01: Removed text that overlapped with [draft-ietf-tls-dtls-connection-id](#)
- 00: Initial version

Appendix B. Acknowledgements

We would like to thank Achim Kraus, Hanno Becker and Manuel Pegourie-Gonnard for their input to this document.

Authors' Addresses

Thomas Fossati
Arm Limited

EMail: thomas.fossati@arm.com

Hannes Tschofenig (editor)
Arm Limited

EMail: hannes.tschofenig@arm.com

