MP-DCCP progress

draft-ietf-tsvwg-multipath-dccp-10

Markus Amend on behalf of the authors, TSVWG @IETF117

Draft status

Gorry‘s early review was incorporated mainly into PR #191 and some subsequent PRs after discussion. All PRs are merged.

Olivier‘s first review available and adressed in the Github Issue tracker. Handshaking procedure optimization required?

Intended RFC status changed from EXP to PS after TSVWG mailinglist discussion: https://mailarchive.ietf.org/arch/msg/tsvwg/R1arXySjvOOwVEoBC7CVQqC-5iU/

Simplified use of the MP-DCCP Linux kernel reference implementation thanks to the new automatic build environment. Any update to the prototype required, for example, after a draft document update, creates a Debian package that facilitates use in Debian and compatible operating systems: https://github.com/telekom/mp-dccp/actions

-08: Added section „Path usage strategies“ draft-ietf-07...draft-ietf-08

-09: Changed document state to PS and incorporated most Gorry‘s review comments. draft-ietf-08...draft-ietf-09

-10: Completed Gorry‘s review: draft-ietf-09...draft-ietf-10

Optimized handshaking procedure 1/3

Raised by Olivier in #225 & #248

MP-DCCP adopted MPTCP handshaking principle for subsequent subflow authentication

However, MPTCP handshaking principle used workaround to cope with limited option header size using key derived tokens.

→ This principle requires a costly collision check before key generation, although MP-DCCP provides a larger header space.

#225: Proposes to use unique Connection Identifier instead of Tokens

#248: Asked for removal of the Key-A information in the final ACK of the initial handshake

Optimized handshaking procedure 2/3

```
 Host A                                         Host B
------------------------                              ----------
Address A1    Address A2                              Address B1
----------    ----------                              ----------
     |             |                                       |
     |             DCCP-Request + MP_CAPABLE               |
     |------- MP_KEY(Key-A(1), Key-A(2),...) ------------->|
     |<---------------------- MP_KEY(Key-B) ---------------|
     |             DCCP-Response +  MP_CAPABLE             |
     |             |                                       |
     |   DCCP-Ack  |                                       |
     |--------- MP_KEY(Key-A) + MP_KEY(Key-B) ------------>|
     |<----------------------------------------------------|
     |   DCCP-Ack  |                                       |
     |             |                                       |
     |             |          DCCP-Request + MP_CAPABLE    |
     |             |--- MP_JOIN(TB,RA) ------------------->|
     |             |<------MP_JOIN(TB,RB) + MP_HMAC(B)-----|
     |             |DCCP-Response  + MP_CAPABLE            |
     |             |                                       |
     |             |DCCP-Ack                               |
     |             |-------- MP_HMAC(A) ------------------>|
     |             |<--------------------------------------|
     |             |DCCP-ACK                               |

MP_HMAC(B) = HMAC-SHA256(Key=d-key(B), Msg=RB+RA)
MP_HMAC(A) = HMAC-SHA256(Key=d-key(A), Msg=RA+RB)
```

So far

```
 Host A                                         Host B
------------------------                              ----------
Address A1    Address A2                              Address B1
----------    ----------                              ----------
     |             |                                       |
     |             DCCP-Request + MP_CAPABLE               |
     |--- MP_KEY(Key-A(1)+CI-A(1), Key-A(2)+CI-A(2),...)-->|
     |<------------------ MP_KEY(Key-B + CI-B) ------------|
     |             DCCP-Response +  MP_CAPABLE             |
     |             |                                       |
     |   DCCP-Ack  |                                       |
     |---------------- MP_KEY(Key-B) --------------------->|
     |<----------------------------------------------------|
     |   DCCP-Ack  |                                       |
     |             |                                       |
     |             |          DCCP-Request + MP_CAPABLE    |
     |             |--- MP_JOIN(CI-B, RA) ---------------->|
     |             |<------MP_JOIN(CI-B,RB) + MP_HMAC(B)---|
     |             |DCCP-Response  + MP_CAPABLE            |
     |             |                                       |
     |             |DCCP-Ack                               |
     |             |-------- MP_HMAC(A) ------------------>|
     |             |<--------------------------------------|
     |             |DCCP-ACK                               |

MP_HMAC(B) = HMAC-SHA256(Key=d-key(B), Msg=RB+RA)
MP_HMAC(A) = HMAC-SHA256(Key=d-key(A), Msg=RA+RB)
```

New Suggested

Optimized handshaking procedure 3/3

Consequence

#225:

- Connection Identifier (CI) is initially exchanged along with the Keys.
- Subsequent subflow establishment use CI in the MP_JOIN request instead of Token

→ MP_KEY needs an additional field to carry CI

→ Draft text describing Token generation and usage needs replacement

#248:

- KEY-A is removed from the final ACK of the inital handshaking as it is an unecessary historical leftover.

→ Change one sentence in the draft.

Does the community objects the following author‘s view?

- **Minimal adaptation** of the handshaking procedure for more efficient implementation and usage
- **Security principle is not changed**

Summary

Remaining issues in Github repo

- Received Olivier‘s first review on mailinglist:
  - Review divided in smaller pieces and added to Github Issue tracker.
  - After first check authors think the issues can be solved in reasonable timeframe
  - Most issues are already commented by the authors

Roadmap:

- We are ready for WGLC assuming that last issues can be solved before next IETF