Multicast Security
-
Quo Vadis?
-

René Struik

Outline

1. Multicast Security:
   - Confidentiality
   - Authentication
   - Replay Protection
2. Suggested approach:
   - Use of DTLS record layer format as is
   - Security processing
   - Compatibility of multicast security with DTLS
3. Changes to:
   - draft-keoh-dice-multicast-security-08
   - draft-kumar-dice-groupcomm-security-00

Multicast Security:

- Confidentiality
- Authentication
- Replay Protection

Multicast Security Objectives

Security services defined in terms of

- Group membership
- Sender of multicast message
- Recipient(s) of multicast message
- Locally maintained status information

Security services:

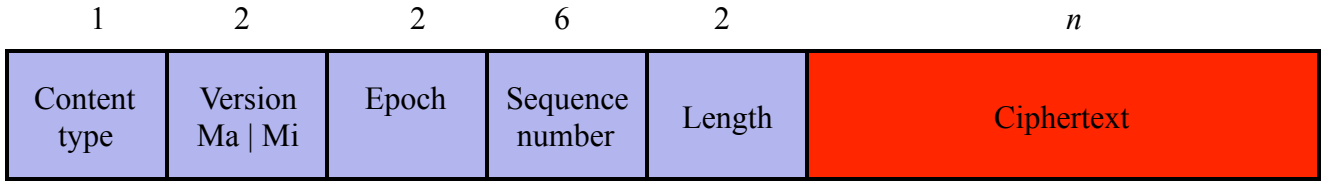
- Confidentiality:
  Assurance that received information is only available to group members
- Authentication:
  - Group authentication: assurance that source of information is group member
  - Source authentication: assurance of precise originator of received information
- Replay protection:
  Assurance that information from specific source is received at most once
- Timeliness: not provided
  Assurance that received information is “relatively fresh” (not “stale”)

Suggested approach:

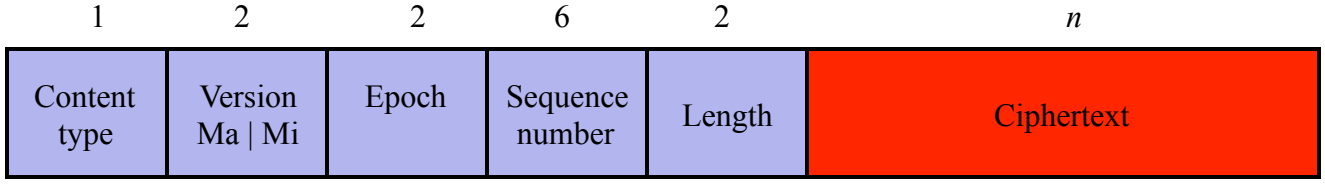
- Reuse of DTLS record layer format as is
- Security processing
- Coexistence with DTLS1.2

Format comparison

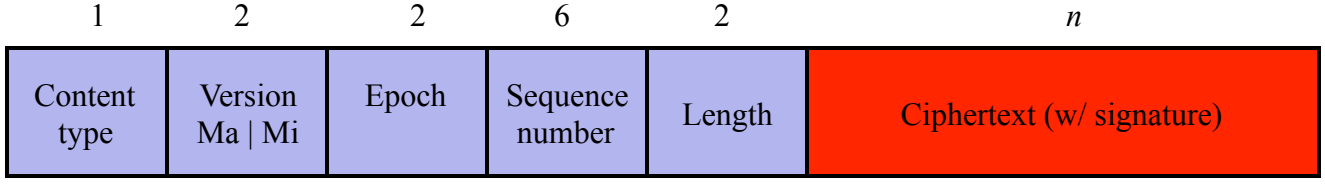
DTLS Record Layer format:



Multicast Security Record Layer format (group authentication):



Multicast Security Record Layer format (source authentication):



Processing comparison (at recipient's end)

DTLS Record Layer:

- Look-up *key* = *G*(*sender, {sender*, *recipient}*, *key id*)
- Apply inverse AEAD cipher (decrypt/check authenticity)

Multicast Security Record Layer format (group authentication):

- Look-up *key* = *G*(*sender, group*, *group key id*)
- Apply inverse AEAD cipher (decrypt/check authenticity)

Multicast Security Record Layer format (source authentication):

- Look-up *key* = *G*(*sender, group*, *group key id*)
- Look-up *signature verification key* = *L*(*sender*, local info)
- Apply inverse AEAD cipher (decrypt/check authenticity)
- Apply signature verification operation (check signature)

Coexistence (at recipient's end)

Device implementing multicast security:

- DTLS traffic received as is (since unicast)
- Multicast traffic received and processed as proposed
- Same AEAD processing

Device implementing DTLS, but *not* implementing multicast security:

- DTLS traffic received as is (since unicast)
- Multicast traffic dropped on the floor

Changes to:

- draft-keoh-dice-multicast-security-08
- draft-kumar-dice-groupcomm-security-00

Changes to draft-keoh-dice-multicast-security-08

Changes:

- Align record format so as to coincide with TLS record format
  - Remove **SenderID** (this re-instates 6-octet **SequenceNumber**)
- Make sure nonce reuse does not occur
  - Use derived key *f*(*group key*, *sender*), rather than using *group key* directly

  NOTE1: here, *group key* is determined from{Multicast IP destination address, port}

  NOTE2: *sender* is originator's IP address (which assumes role of **SenderID**)
- Define local logic for key look-up
  - Look-up *group key* = *G*(*sender, group*, *group key id*)

  NOTE: with DTLS, *key* = *G*(*sender, key id*) = *G'*(*sender*, *recipient*, *key id*),

  So one can use uniform local key look-up

NOTE:

- No changes to replay protection (via comparison of **SequenceNumber** of packet and locally maintained status information) required

Changes to draft-kumar-dice-groupcomm-security-00

Changes:

- Align record format so as to coincide with TLS record format
  - Use same format as suggested with draft-keoh-dice-multicast-security-08
- Make sure nonce reuse does not occur
  - Use derived key *f*(*group key*, *sender*), rather than using *group key* directly

  NOTE1: here, *group key* is determined from{Multicast IP destination address, port}

  NOTE2: *sender* is originator's IP address (which assumes role of **SenderID**)
- Define local logic for key look-up
  - Look-up *group key* = *G*(*sender*, *group*, *group key id*)

  NOTE: with DTLS, *key* = *G*(*sender*, *key id*) = *G'*(*sender*, *recipient*, *key id*),

  So one can use uniform local key look-up
- Define local logic for signature look-up
  - Look-up *signature verification key* = *L*(*sender*, local info)

NOTE:

- No changes to replay protection (via comparison of **SequenceNumber** of packet and locally maintained status information) required

Next Steps

Write new draft, borrowing from current discussion