IKE_INTERMEDIATE Exchange Update

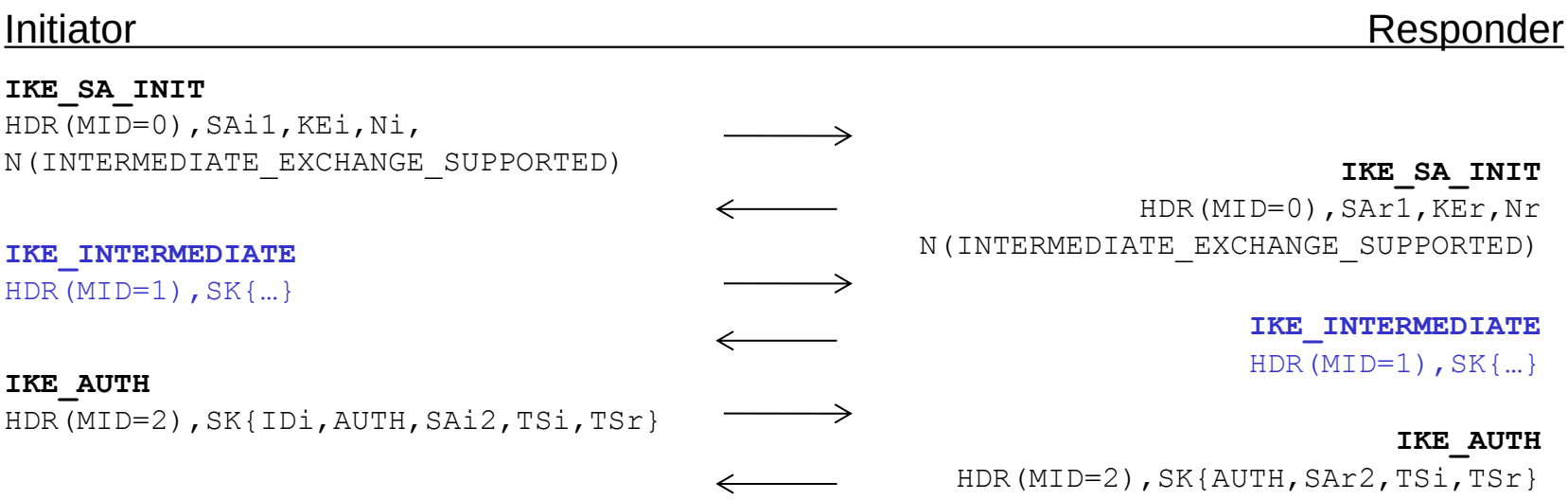
draft-ietf-ipsecme-ikev2-intermediate-01

Valery Smyslov
svan@elvis.ru

IETF 105

Exchange Overview

One or more IKE_INTERMEDIATE (ex IKE_AUX) Exchanges may take place between IKE_SA_INIT and IKE_AUTH. Their use is negotiated via exchange of `INTERMEDIATE_EXCHANGE_SUPPORTED` notifications:



The exchanges can be used to transfer large amount of data prior IKE_AUTH (e.g. QSKE public keys), since standard IKE Fragmentation works for them

Changes from draft-smyslov-ipsecme-ikev2-aux-02

- Exchange is renamed from INTERMEDIATE to IKE_INTERMEDIATE
- Considerations for integration with IKE Session Resumption are added
  - support for IKE_INTERMEDIATE must be re-negotiated in case of IKE session resumption

IKE_INTERMEDIATE Exchange Authentication

- All IKE_INTERMEDIATE messages are included into the AUTH payload calculation by each party

```
InitiatorSignedOctets = RealMessage1 | NonceRData | MACedIDForI [| IntAuth]
ResponderSignedOctets = RealMessage2 | NonceIData | MACedIDForR [| IntAuth]

IntAuth = IntAuth_1 | [| IntAuth_2 [| IntAuth_3]] ...

IntAuth_n = IntAuth_n_I | IntAuth_n_R
IntAuth_n_I = prf(SK_pi_n, [IntMessage_n_I_P |] IntMessage_n_I_A)
IntAuth_n_R = prf(SK_pr_n, [IntMessage_n_R_P |] IntMessage_n_R_A)
```

IntMessage_n_[I/R]_P – content of Encrypted payload before encryption and possible fragmentation (not including payload header, IV, ICV, Pad Length and Padding)

IntMessage_n_[I/R]_A – part of the message from the beginning of IKE header till the end of Encrypted payload header

Current Authentication Scheme

- For the purpose of authentication IKE_INTERMEDIATE messages are always treated as if they were sent unfragmented, regardless of the actual form they were sent in
  - this allows IKE_INTERMEDIATE to be fully compliant with RFC7383
  - In case of IKE fragmentation IKE header and Encrypted payload header are reconstructed
- The input to prf includes (possibly reconstructed) IKE header and Encrypted Payload header
  - Length fields in these headers count IV length, ICV length and padding length, despite the fact. that these fields themselves are not included into AUTH calculation

Issues with the Current Scheme

- Reconstructing headers is inconvenient for some implementations
  - it requires to know the size of IV, ICV, padding, that may be unavailable at the point prf is computed in highly modular implementations
- The peers may disagree on the length field in the headers, e.g. by applying padding of different size
  - authentication may fail even when message itself is unmodified

Proposed Changes

- For the purpose of prf calculating modify Length fields in IKE header and Encrypted Payload header so that they don't count IV length, ICV length and padding length (along with Pad Length field)
  - thus these fields will only count size of data that is input to prf
- Change the order of inputs to prf from

```
IntAuth_n_I = prf(SK_pi_n, [IntMessage_n_I_P |] IntMessage_n_I_A)
IntAuth_n_R = prf(SK_pr_n, [IntMessage_n_R_P |] IntMessage_n_R_A)
```

to more natural

```
IntAuth_n_I = prf(SK_pi_n, IntMessage_n_I_A [| IntMessage_n_I_P])
IntAuth_n_R = prf(SK_pr_n, IntMessage_n_R_A [| IntMessage_n_R_P])
```

  - the reverse order is no more needed since we do know all the values for length fields in the headers before calling encryption function

Thank you!

- Comments?
- Questions?
- Way forward?