Extended IKEv2 Payload Format

draft-smyslov-ipsecme-ikev2-extended-pld

Valery Smyslov
svan@elvis.ru

IETF 116

Problems with Existing Format

- Payload Length field occupies 2 bytes, so payload size is limited to 64 Kbytes
  - might not be enough for some PQ algorithms
  - no problem with Message size, which is limited to 4 Gbytes
- Many payloads contain substantial redundancy
  - Payload Length field occupies 2 bytes, while most payloads are shorter
  - most parameters occupy 2 bytes, while less than 256 values are defined
  - zero-filled `RESERVED` fields

Existing Proposals

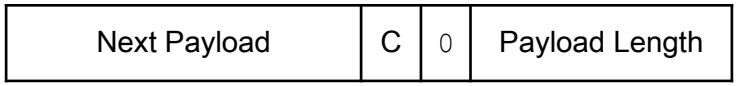
- A Larger Internet Key Exchange version 2 (IKEv2) Payload
  draft-nir-ipsecme-big-payload
- Beyond 64KB Limit of IKEv2 Payloads
  draft-tjhai-ikev2-beyond-64k-limit
- Compact Format of IKEv2 Payloads (expired)
  draft-smyslov-ipsecme-ikev2-compact
  (expired)

Extended Payload Format Overview

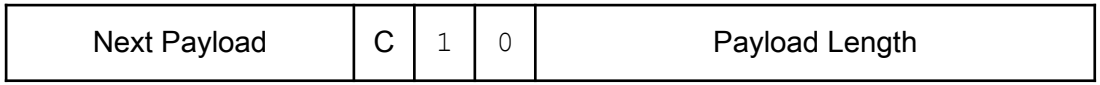
- Three formats for new Generic Payload Header
  - for small payloads (up to 64 bytes)
  - for medium size payloads (up to 8 Kbytes)
  - for large payloads (up to 512 Mbytes)
- No `RESERVED` fields
- Revise some existing payloads headers to reduce their size
  - remove unnecessary fields
- Special Format for some payloads (SA, some status notifies)

Extended Generic Payload Header Format

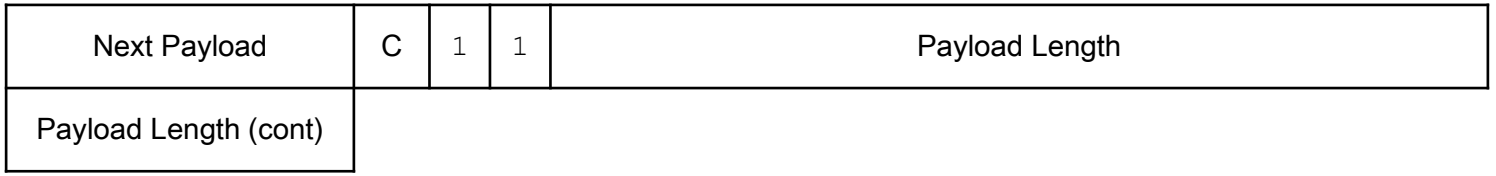
1. Small payloads (2 bytes, 6 bits for Payload Length)



2. Medium size payloads (3 bytes, 13 bits for Payload Length)



3. Large payloads (5 bytes, 29 bits for Payload Length)



Revise some Payload Headers

The following payload headers are revised:

- Key Exchange Payload
  - no `RESERVED` field (2 bytes)
- Identification, Authentication, Configuration Payloads
  - no `RESERVED` field (3 bytes)
- Traffic Selector Payload
  - no `RESERVED` field (3 bytes)
  - no `Number of TSs` field (1 byte)
- Traffic Selector
  - no `Selector Length` field (2 bytes)

Special Format for some Payloads

Special format for:

- SA Payload
  - SA Payload grows quickly as more and more new transforms are defined and offered by initiators
- Notify Payload with some Status Type Notification containing no data
  - Exchange of such payloads is a common way to negotiate support for various protocol extensions, so initial IKEv2 messages grow up as more and more extensions are defined

Both payloads contain a lot of redundancy and can be effectively compacted.

SA Payload

- No `RESERVED` fields
- No generic header in `Proposal` substructure
- Encode `Transform` substructure as variable-length structures

Transform Encoding

1-byte: for `Encryption`, `Key Exchange`, `PRF`, `ESN` Transform Types for limited number of Transform IDs

1-byte: for some future Transform Types (e.g. for G-IKEv2) and limited number of Transform IDs

2-bytes: for `Additional Key Exchange` Transform Types and for other Transform Types with Transform IDs that don't fit into 1-byte encoding

3-bytes: for Transform IDs that don't fit into 1-byte and 2-bytes encodings

5-512 bytes: for remaining Transform IDs or in case there are Attributes (other than `Key Length`)

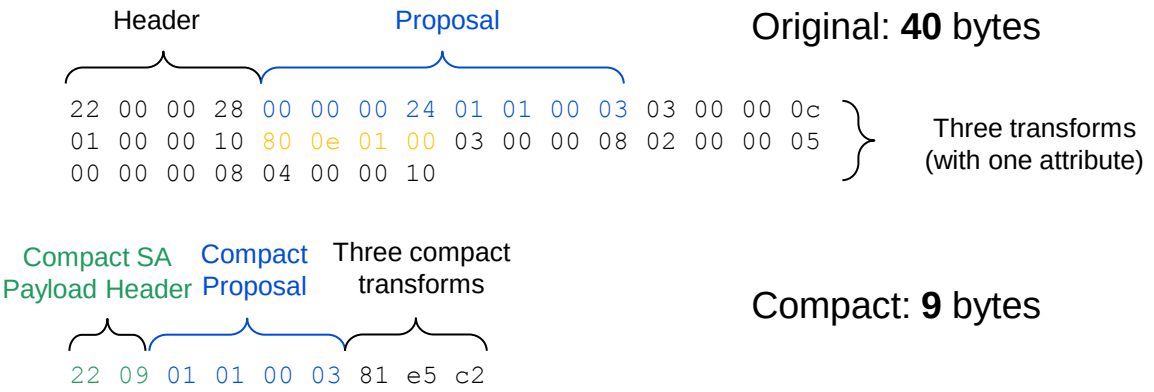
Transform Encoding Summary

Name	Format	Length	Transform Types	Transform IDs
Short	0**ttt**vvvv	1	13-20	0-15
Encryption (128)	100vvvvv	1	1	11-42
Encryption (256)	101vvvvv	1	1	11-42
KE	110vvvvv	1	4	0, 14-44
PRF	1110vvvv	1	2	2-15
ESN	111110vv	1	5	0-3
Long 1	11110**ttt tt**vvvvvv	2	1-31	0-63
Long 2	1111110**t tttt**vvvv vvvvvvvv	3	1-31	0-4095
Full	1111111**t ttttttt***l lllllll* vvvvvvvv vvvvvvvv	5 up to 512 (in case of attributes)	any	any

Example of Compact SA Payload

SA Payload with one Proposal and three Transforms:

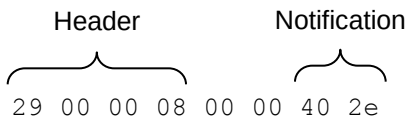
- `ENCR_AES_CCM_16` (256 bits key)
- `PRF_HMAC_SHA2_256`
- `4096-bit MODP Group`



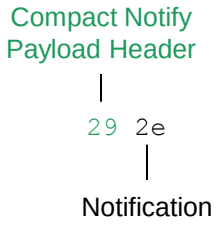
Notify Payload

Outline: encode notification in one octet (limited to first 256 status notifications) and omit all other fields from Notify Payload

Example: Notify Payload with `IKEV2_FRAGMENTATION_SUPPORTED` notification.



Original: **8** bytes



Compact: **2** bytes

Negotiation

If new format is used from the very beginning then the following options exist

- New status notify `EXTENDED_PAYLOAD_FORMAT`
  - extended format cannot be used in `IKE_SA_INIT`
  - suitable if only large payloads are needed
- New initial exchange `X_IKE_SA_INIT`
  - functionally equivalent to `IKE_SA_INIT`, but may contain payloads in extended (compact) format
  - old responders would return `INVALID_SYNTAX` notify

Transport Issues

Transport issues for transferring large payloads (> 64 Kbytes) are out of scope. Possible solutions:

- IKE over TCP combined with IKE fragmentation (to solve limitation on 64 Kbytes on a single IKE message over TCP)
- Mixed Mode (defined in draft-tjhai-ikev2-beyond-64k-limit: IKE over TCP + IKE fragmentation combined with plain ESP or ESP over UDP) can be used to avoid ESP performance degradation when used with TCP encapsulation

Discussion

- Get rid of `SPI Size` in Proposal substructure, Delete and Notify payloads (can be deducted from `Protocol ID`)?
- Get rid of `Proposal Num` in Proposal substructure?
- New Payload types or reuse existing types?
  - 9+ new payload formats – too many?
- Certificates consume a lot of space, can be compressed (out of scope)
  - RFC 8879 is an example of certificate compression

Thank you!

- Comments?
- Questions?
- Any interest in this work?