New IKEv2 Payload Format

Valery Smyslov
svan@elvis.ru

IETF 110

Existing Format Redundancy

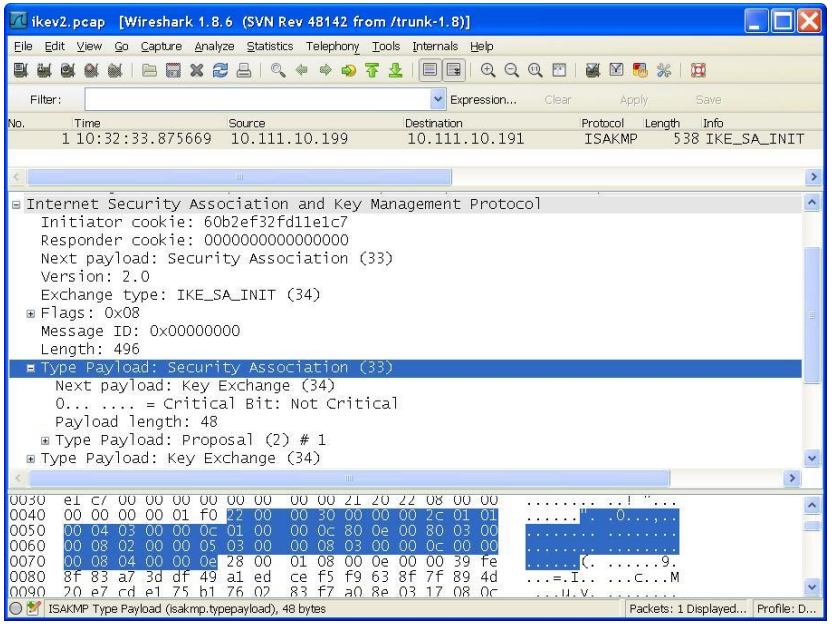
Many payloads contain substantial redundancy

- 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

Example: SA Payload on the right contains one Proposal with four Transforms:

- `ENCR_AES_CBC` (128 bits)
- `PRF_HMAC_SHA2_256`
- `AUTH_HMAC_SHA2_256_128`
- `2048-bit MODP Group`

Payload size is **48** bytes, among which **24** bytes are zeroes.



Existing Format Limitations

- Payload Length field occupies 2 bytes, so payload size is limited to 64 Kbytes
  - no problem with Message size, which is limited to 4 Gbytes

Making Payloads Smaller

- Would decrease power and network bandwidth consumption (important for IoT devices)
- Would decrease chances of IP fragmentation in the IKE_SA_INIT and IKE fragmentation in the rest exchanges

Lifting 64 Kbytes Size Limit

- Would allow using PQ algorithms with long public keys and signatures
  - draft-tjhai-ikev2-beyond-64k-limit
- Would allow transferring large chunks of data (e.g. in CP payload)

New Format Requirements

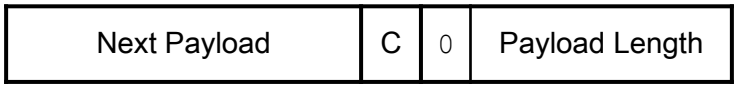
- Must be suitable for both small and large payloads
- Must be applicable to any payload type, including not yet defined ones
  - some payloads may have special format if it is justified
- The encoder/parser must remain simple and consume low resources

New Format Proposal

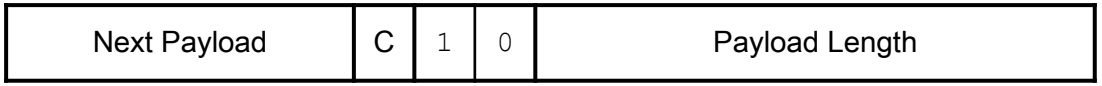
- Three possible formats for new Generic Payload Header
  - for small payloads (up to 64 bytes)
  - for medium size payloads (up to 8 Kbytes)
  - for large payloads (up to 512 Mbytes)
- No RESERVED fields
- Revise existing payloads headers to reduce their size
  - remove unnecessary fields
- Special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

New Generic Payload Header

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)

Revised Existing Payload Headers

The following payload headers can be revised:

- Key Exchange, Identification, Authentication, Configuration
  - remove `RESERVED` field
- Notify
  - remove `SPI Size` field (can be deducted from Protocol ID)
- Delete
  - remove `SPI Size` field (can be deducted from Protocol ID)
  - remove `Num of SPIs` field (can be deducted from Payload Length)
- Traffic Selector
  - remove `RESERVED` field
  - remove `Number of TSs` field (can be deducted from Payload Length)

Special Format

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 and 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.

(*) Inspired by draft-smyslov-ipsecme-ikev2-compact

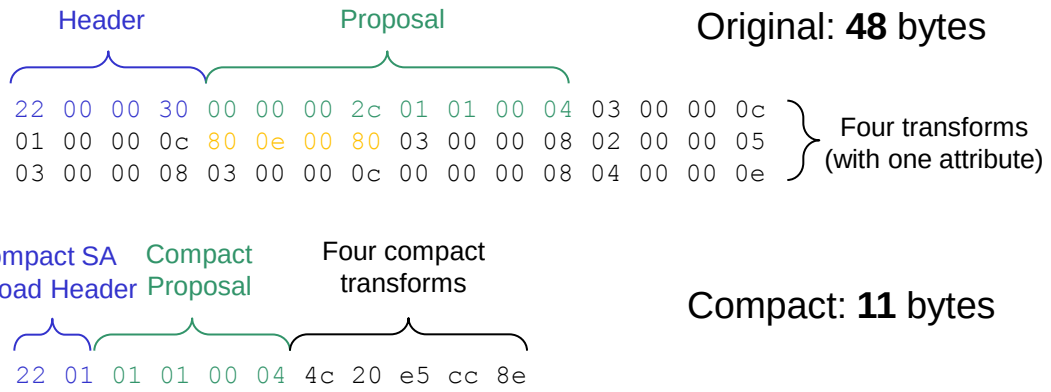
SA Payload

Outline:

- Remove all `RESERVED` fields
- Remove `Length` fields in substructures (where they are unnecessary)
- Encode all currently defined transforms w/o attributes using one octet (both `Transform Type` and `Transform ID`)
- Encode currently defined `Encryption` transforms having `Key Length` attribute using two octets
- Leave possibility to encode arbitrary (even not yet defined) `Transform Type` and `Transform ID`, as with regular format

Example: SA Payload with one Proposal and four Transforms:

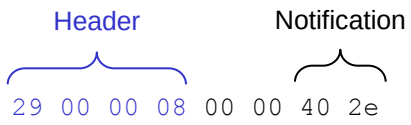
- `ENCR_AES_CBC` (128 bits)
- `PRF_HMAC_SHA2_256`
- `AUTH_HMAC_SHA2_256_128`
- `2048-bit MODP Group`



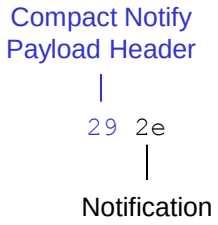
Notify Payload

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 major IKE version (v3)
  - old responders would return `INVALID_MAJOR_VERSION`
- New type of initial exchange (e.g. `ALT_IKE_SA_INIT`)
  - old responders would return `INVALID_SYNTAX`
- New critical payload in the `IKE_SA_INIT`, followed by payloads in new format
  - old responders would return `UNSUPPORTED_CRITICAL_PAYLOAD`

Discussion

- We don’t need to assign new payload types except for special format payloads (SA and empty status Notify), do we? What about revised payloads?
- Transport issues for transferring large payloads are out of scope
  - IKE over TCP combined with IKE fragmentation (to solve limitation on 64 Kbytes on a single IKE message over TCP) can be used
  - do we need mixed mode – IKE over TCP combined with plain ESP or ESP over UDP?
- Certificates consume a lot of space, can be compressed
  - RFC 8879 is an example of certificate compression
  - in some use cases draft-mattsson-cose-cbor-cert-compress can be used

Thanks

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