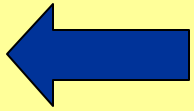
Group Domain of Interpretation

<draft-ietf-msec-gdoi-00.txt>

Mark Baugher (Cisco)
Thomas Hardjono (Nortel)
Hugh Harney (SPARTA)
Brian Weis (Cisco)

Group DOI

GDOI Design Criteria



Overview of related protocols

GDOI Protocol Overview

Current Status

Future Plans

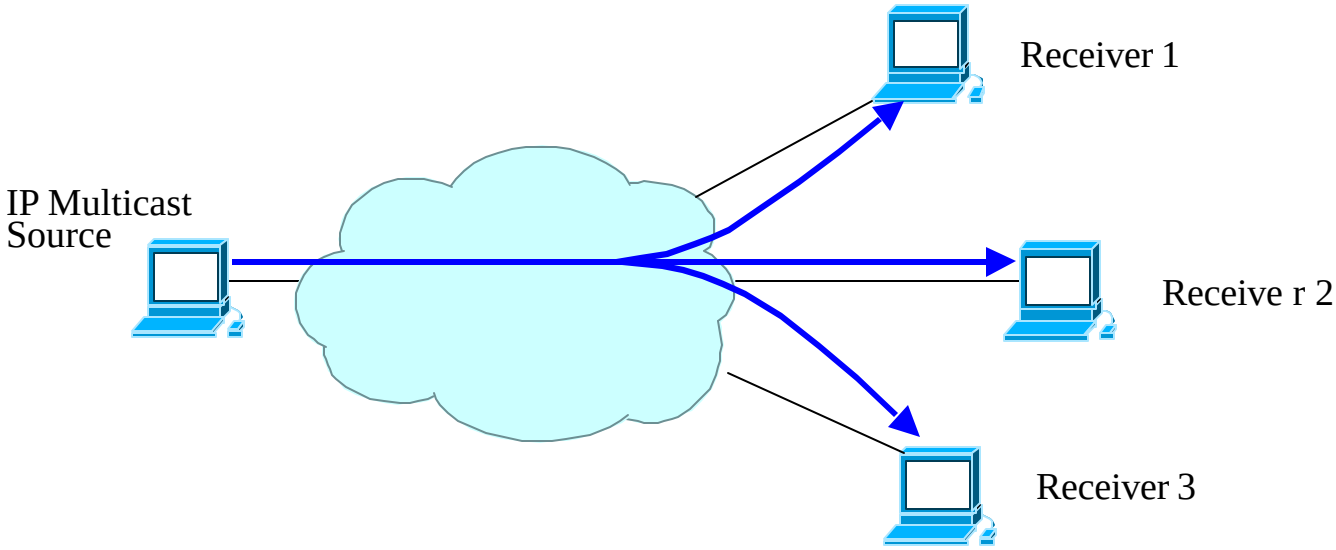
Design criteria for GDOI

- General enough for keying many applications
- Co-existence with unicast key management
- Make use of existing key management protocol definitions when possible.

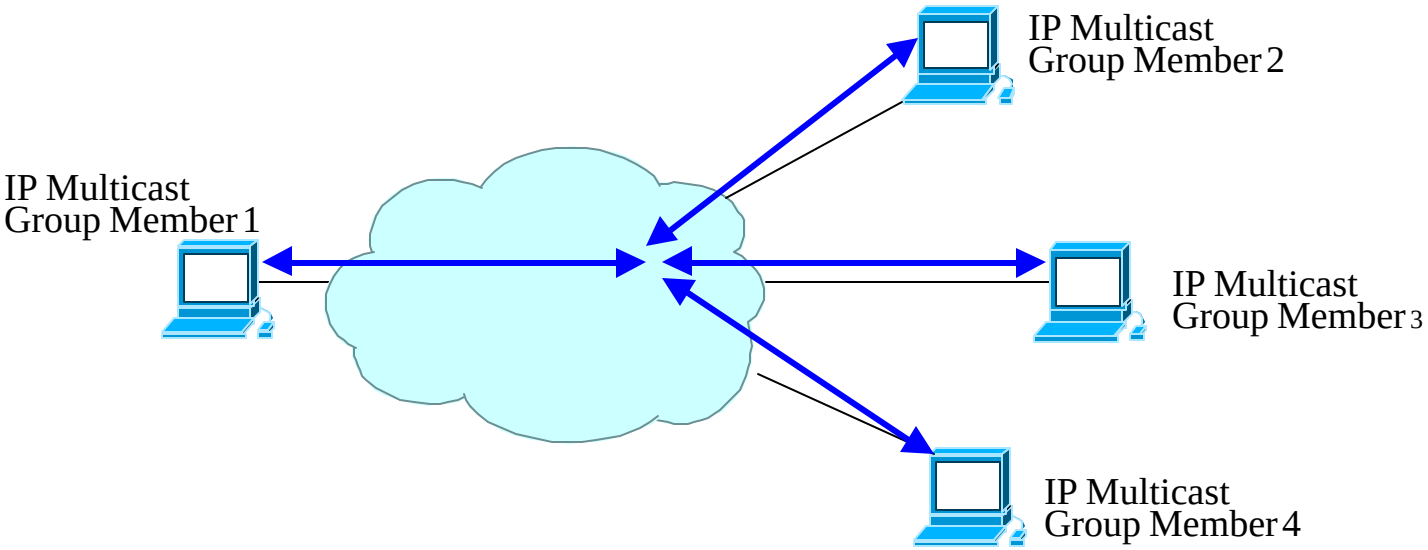
Applications

- GDOI is designed to handle many application scenarios
- Each of the applications has unique characteristics for key management

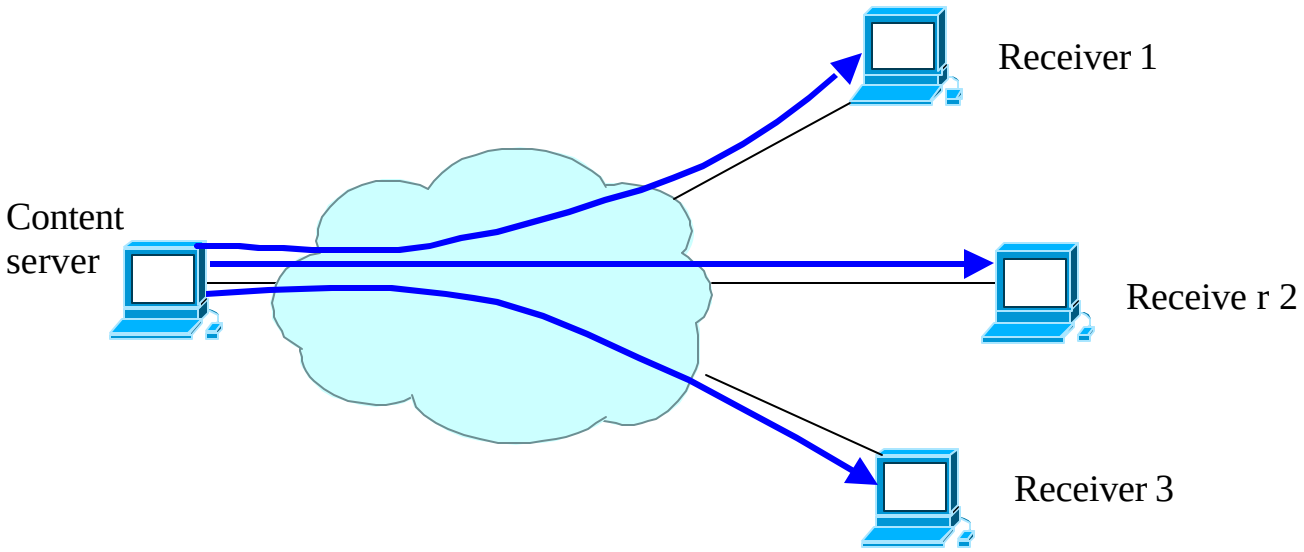
Single-source multicast



Multiple-source multicast



On-demand content distribution



Co-existence with unicast key management

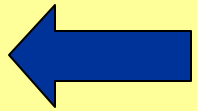
- There will be a need for systems to protect both group and unicast traffic.
- Having two independent key mgmt frameworks on one system is unnecessarily complex.
  - Two independent key mgmt frameworks for IPSEC (one unicast, one multicast) is unnecessary.

Co-existence with IKE

- Many systems have implemented ISAKMP and IKE for unicast key mgmt.
  - PCs and Workstations
  - Firewalls, Gateways, VPN boxes
- We believe that these devices can be used for group key mgmt.

Group DOI

GDOI Design Criteria

Overview of related protocols 

GDOI Protocol Overview

Current Status

Future Plans

ISAKMP

- ISAKMP [RFC 2408] specifies a general framework for key management
  - It defines payloads for authentication, policy negotiation, and key generation.
  - It defines basic key management exchanges.

Domains of Interpretation

- ISAKMP provides for multiple domains of interpretation (DOIs)
- One DOI is defined for IPSEC
- Other recently proposed ISAKMP DOIs:
  - `draft-tsenevir-smpls-doi-00.txt`
  - `draft-arkko-map-doi-01.txt`
  - `draft-lordello-ipsec-vpn-doi-00.txt`

IPSEC DOI & IKE

- IPSEC DOI [RFC 2407] defines IPSEC-specific payloads and policy definitions
- IKE [RFC 2409] defines exchanges in two phases:
  - IKE Phase 1 sets up a “secure channel” between IKE peers.
  - IKE Phase 2 negotiates specific IPSEC policy.

IKE Phase 1

- Authenticates the encryption peer.
- Negotiates “secure channel” policy for the key management session.

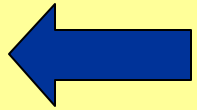
IKE Phase 1 services

- IKE Phase 1 provides the following security services for subsequent exchanges:
  - *Confidentiality* -- encryption of exchange payloads
  - *Integrity* -- HASH payload provides an HMAC
  - *Replay protection* -- NONCE payload provides liveliness proof

Group DOI

GDOI Design Criteria

Overview of related protocols

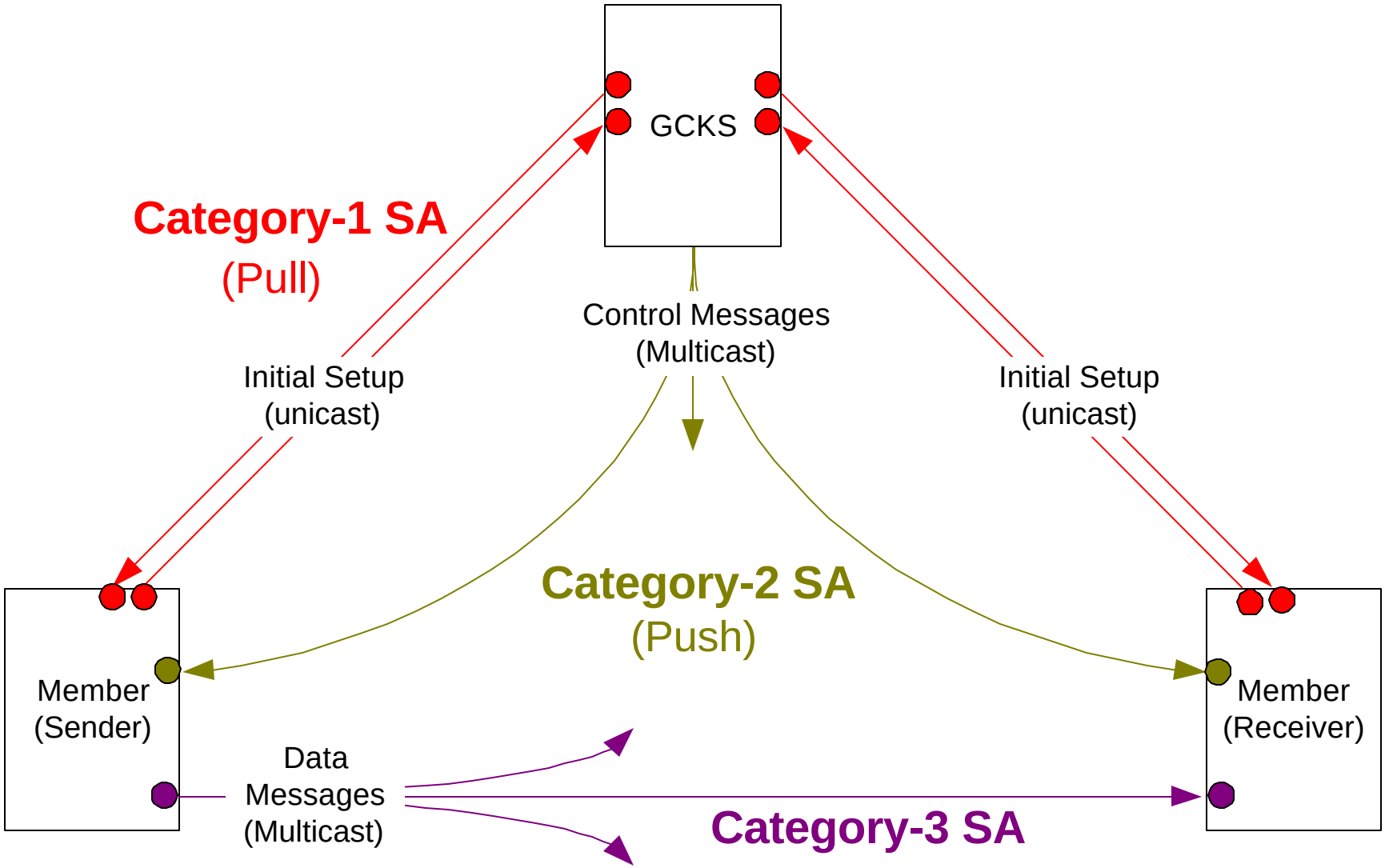
GDOI Protocol Overview 

Current Status

Future Plans

Group DOI Overview

- Proposes that a new DOI number be assigned for GDOI.
- Defines a new Category-1 SA (“Pull”) exchange for initial group key mgmt.
- Defines a Category-2 SA (“Push”) exchange for subsequent key updates. This exchange can be multicasted for efficiency.


GCKS
Category-1 SA
(Pull)
Control Messages
(Multicast)
Initial Setup
(unicast)
Initial Setup
(unicast)
Category-2 SA
(Push)
Member
(Sender)
Member
(Receiver)
Data
Messages
(Multicast)
Category-3 SA

“Pull Protocol”

- Group member “pulls” keys over the IKE Phase 1 “secure channel”.
- Four message exchange between group member and GCKS.
- When complete the group member has all keys necessary to decrypt and authenticate:
  - data packets
  - “push” key management messages

Message 1: Request

```
Initiator (Member)                  Responder (GCKS)
------------------                  ----------------
HDR*, HASH(1), Ni, ID      -->

* Protected by IKE Phase 1 SA Hashes, encryption occurs after HDR
```

```
  HASH(1) = prf(SKEYID_a, M-ID | Ni | ID)
```

- HASH provides message authentication
- NONCE is used for replay protection
- ID indicates the desired group to join

Message 2: Policy Push

```
Initiator (Member)                    Responder (GCKS)
------------------                    ----------------
                            <--       HDR*, HASH(2), Nr, SA
```

```
  HASH(2) = prf(SKEYID_a, M-ID | Ni_b | Nr | SA)
```

- SA contains specific policy for the Category-2 and Category-3 SAs. E.g., which crypto algorithms to use.

Message 3: Ack

```
Initiator (Member)                  Responder (GCKS)
------------------                  ----------------
HDR*, HASH(3) [, KE_I]     -->

          [,CERT] [,POP_I]
```

```
  HASH(3) = prf(SKEYID_a, M-ID | Ni_b | Nr_b [ | KE_I ] [| POP_I])
```

- KE_I obtains perfect forward secrecy (if desired)
- CERT send a public key used for authorization (if needed for POP_I)
- POP_I provides evidence that the client has possession of a private or secret key

Message 4: Key Download

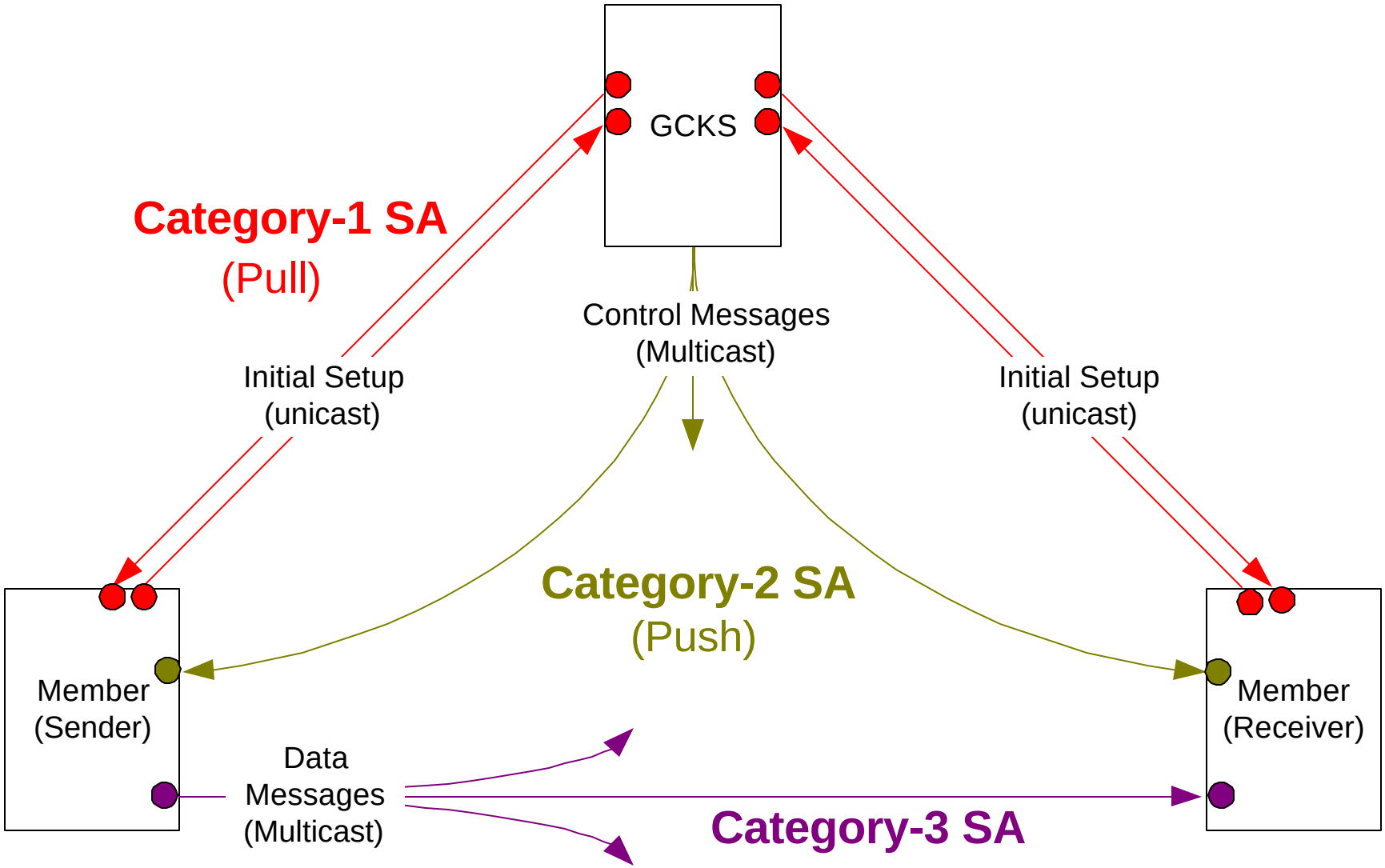
```
Initiator (Member)                    Responder (GCKS)
------------------                    ----------------
                          <--          HDR*, HASH(4), [KE_R,] SEQ, KD
                                                        [,CERT] [,POP_R]
```

```
  HASH(4) = prf(SKEYID_a, M-ID | Ni_b | Nr_b [ | KE_R ] | SEQ | KD [| POP_R])
```

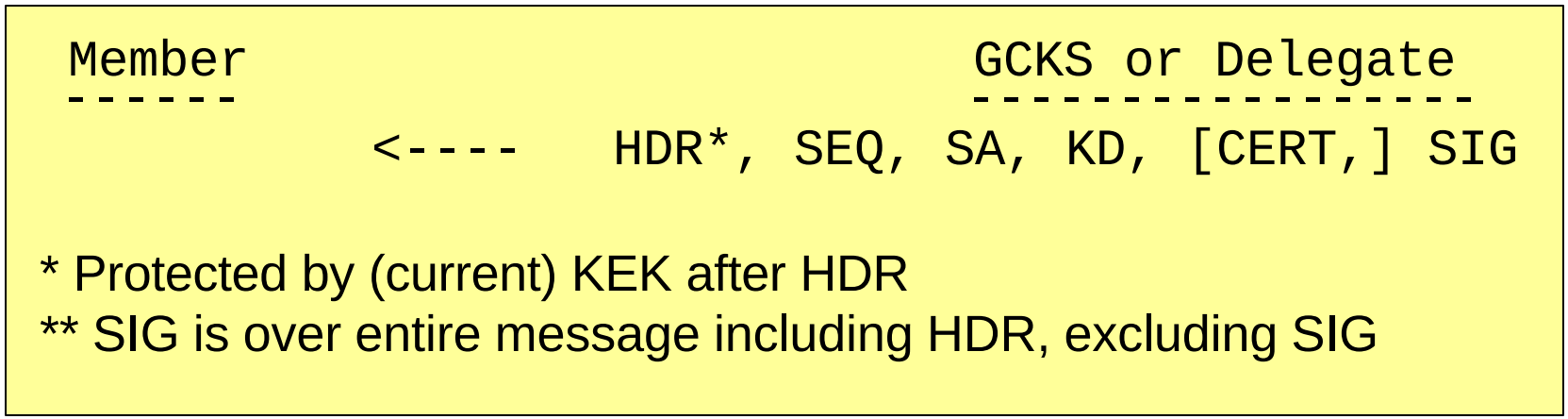
- SEQ provides the sequence number which will be used for the next “push” message.
- KD provides the keys for the policy delivered in the SA payload

Entire “Pull” Protocol

```
Initiator (Member)                    Responder (GCKS)
-----------------                     ----------------

HDR*, HASH(1), Ni, ID       -->

                            <--       HDR*, HASH(2), Nr, SA

HDR*, HASH(3) [, KE_I]      -->
          [,CERT] [,POP_I]

                            <--       HDR*, HASH(4), [KE_R,] SEQ, KD
                                                    [,CERT] [,POP_R]

* Protected by IKE Phase 1 SA Hashes, encryption occurs after HDR
```


GCKS
Category-1 SA
(Pull)
Control Messages
(Multicast)
Initial Setup
(unicast)
Initial Setup
(unicast)
Category-2 SA
(Push)
Member
(Sender)
Member
(Receiver)
Data
Messages
(Multicast)
Category-3 SA

“Push” Message



- The “cookie pair” in the ISAKMP HDR acts as a SPI which identifies the group.
- SEQ contains a counter used for replay protection
- SIG contains a digital signature of the packet for authentication

Payloads

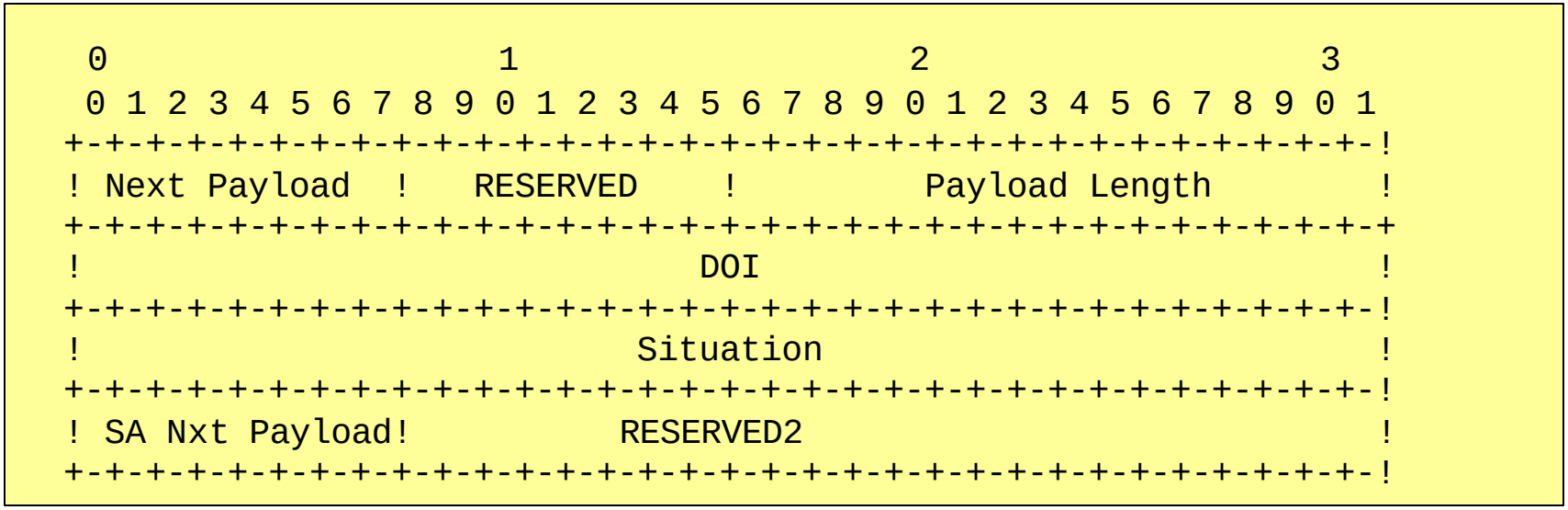
Modified Payload:

- SA (further specified for GDOI DOI)

New Payloads:

- POP payload is defined
- SEQ payload is defined
- KD payload is defined

SA Payload



- DOI: Need an IANA assigned value
- Situation: must be zero in this version

SA Payloads

SA_KEK provides policy for the Category-2 SA

– There may be 0 or 1 present

- SA_TEK provides policy for the Category-3 SA

–There may be 0 - n present

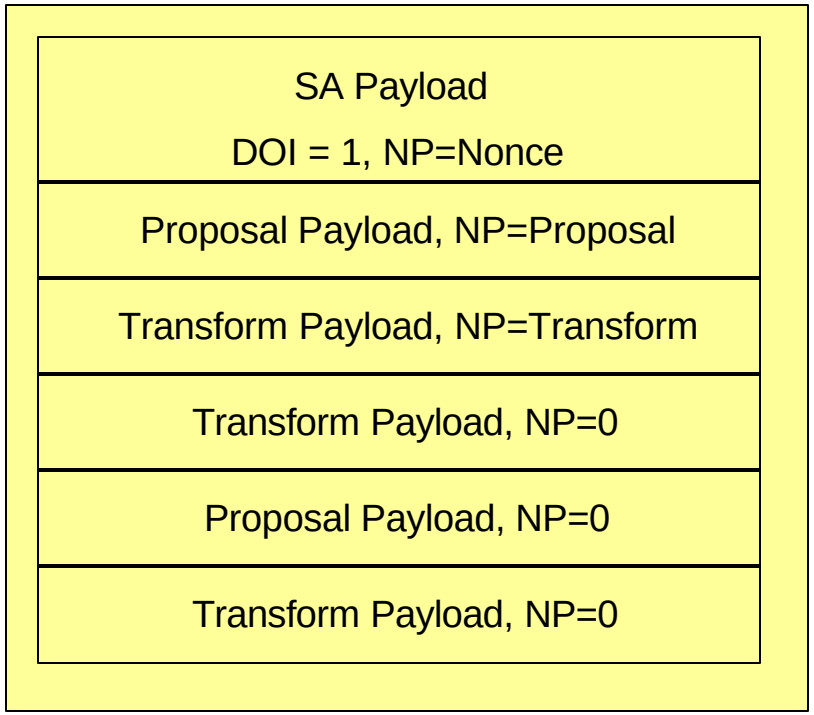
At least one payload must be present!

SA Payload Usage Examples

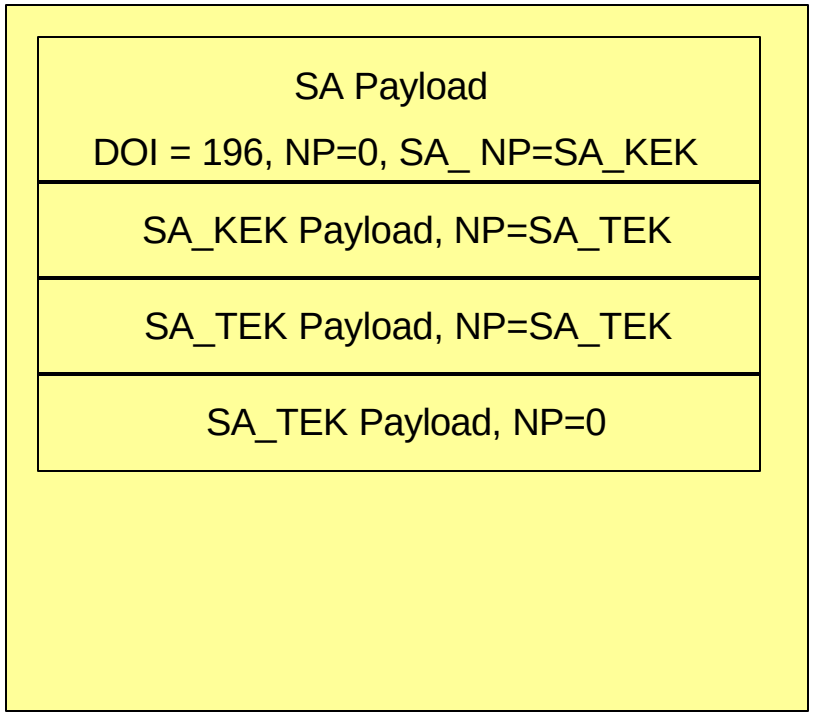
- A multicast application could include SA_KEK and SA_TEKs in the pull and push messages.
- An on-demand content distribution application could include an SA_KEK in the pull message, and send only SA_TEKs in the push message.

SA Payload linking

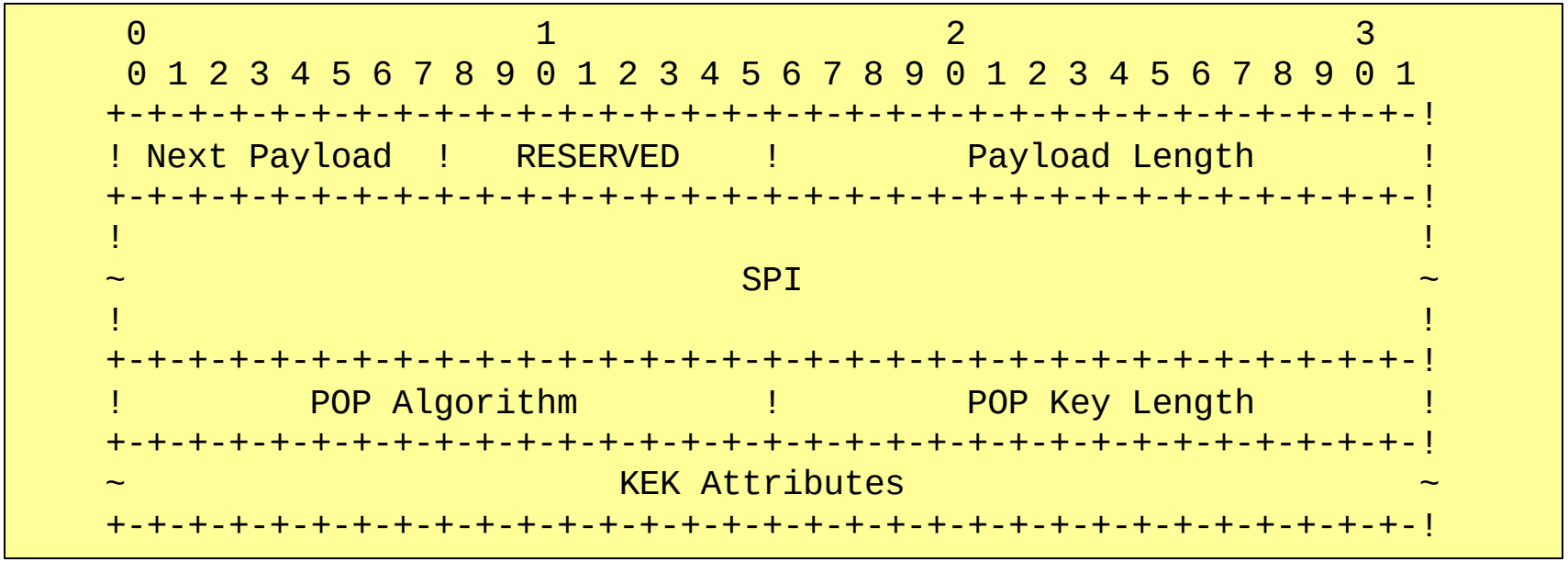
IPSec SA Payload example



GDOI SA Payload example

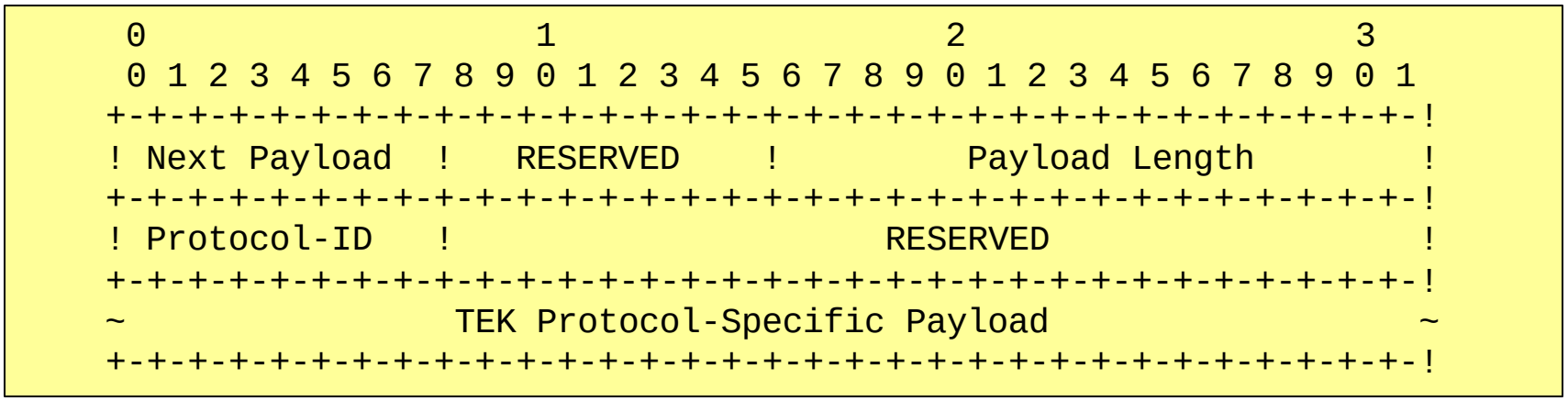


SA_KEK Payload



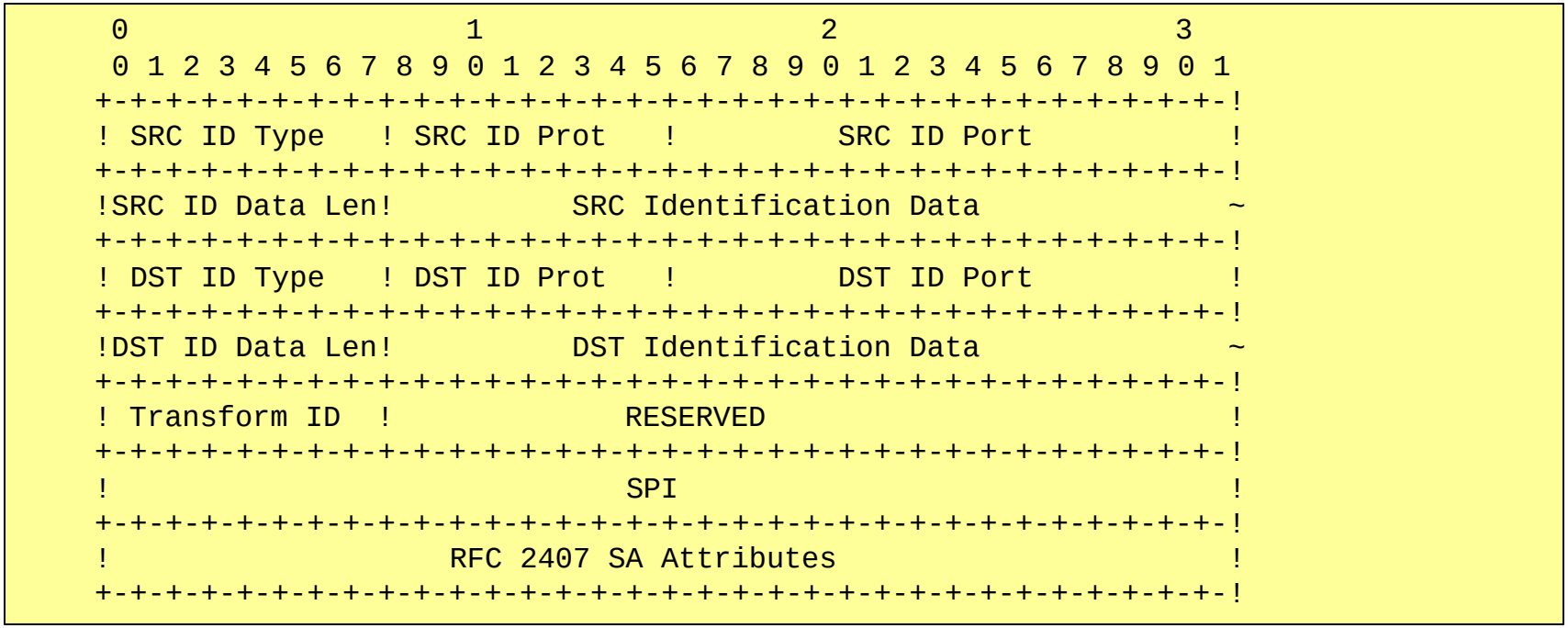
- SPI: ISAKMP initiator/responder cookies for push msg
- POP Algorithm: Proof of Possession algorithm type
- KEK Attributes: algorithm and keying policy for push msg

SA_TEK Payload



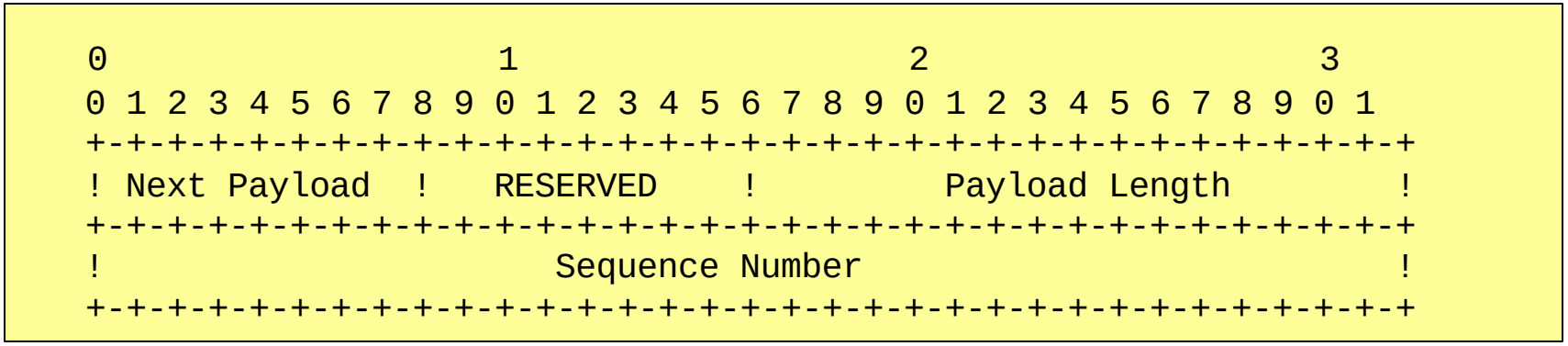
- Protocol-ID: TEK-specific payload type
  - ESP
  - MESP
  - AMESP

ESP SA_TEK Payload



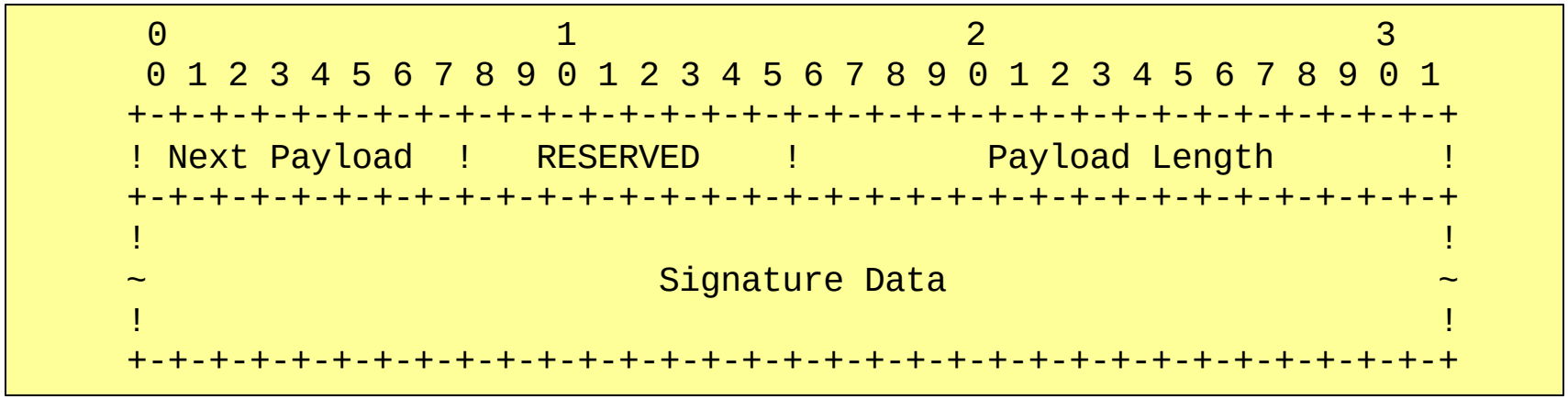
- SRC ID/DST ID: Similar to IPSEC ID payloads
- Transform ID: Defines crypto transform (e.g., ESP 3DES)
- SPI: ESP SPI

SEQ Payload



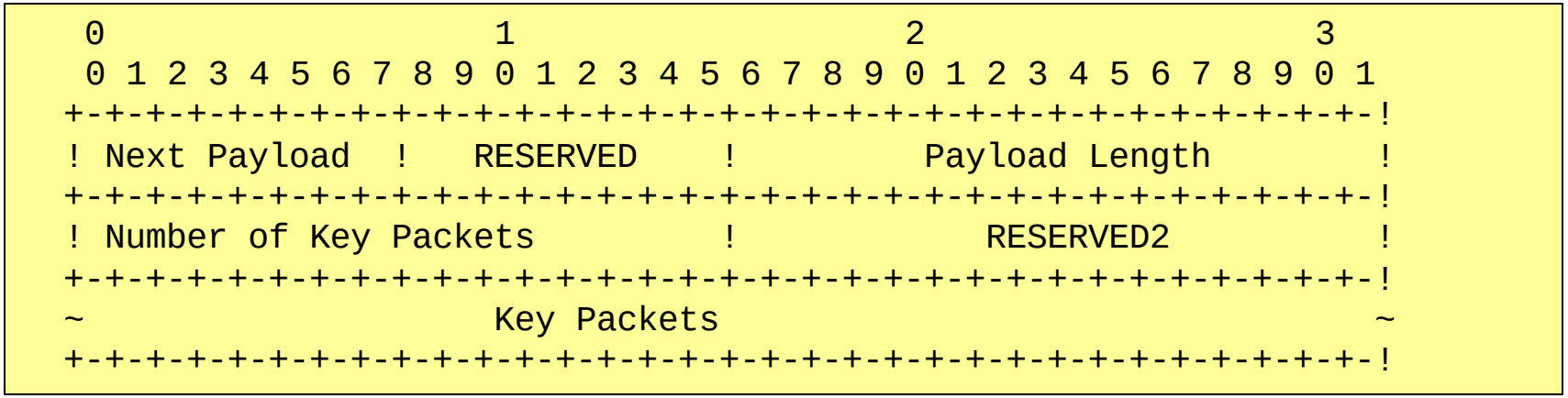
- Contains a sequence number used for replay protection
- Initializes the group members replay window when included in a “pull” message.
- Acts as a replay protection when included in a “push” message.

POP Payload



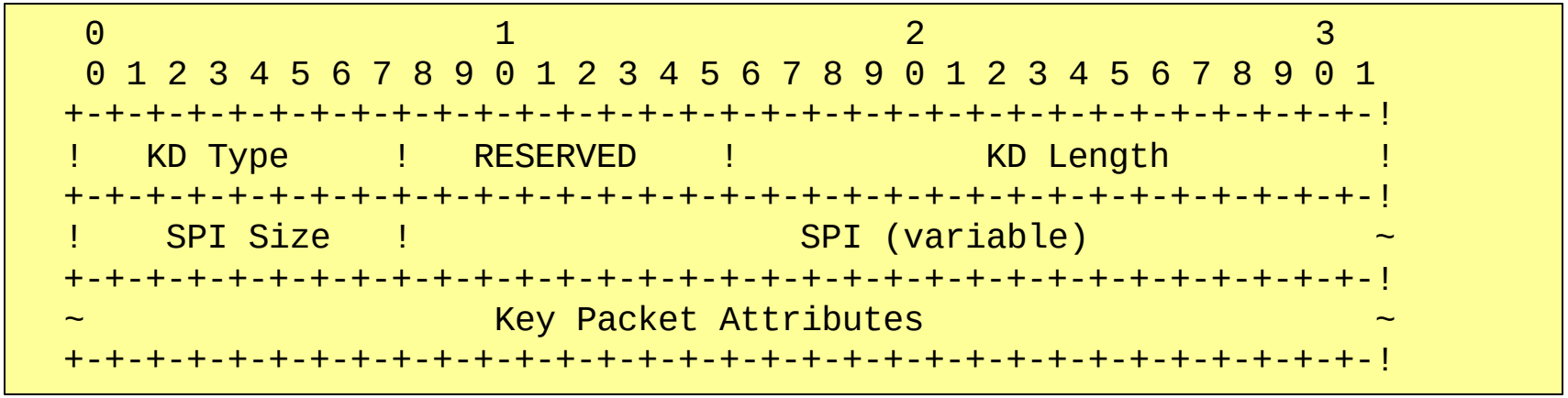
- Identical to a SIG payload, but uniquely named for clarification
- Signature in the POP payload provides evidence that group member is in possession of the private key
- Signature is over entire payload, excluding SIG

KD Payload



- KD is a header to the actual key packets.

Key Download Packet



- KD Type: TEK, KEK (single key), LKH (key array)
- SPI: Matches a SPI sent in an SA_KEK or SA_TEK payload
- Attributes: Encryption keys

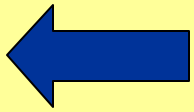
Group DOI

GDOI Design Criteria

Overview of related protocols

GDOI Protocol Overview

Current Status



Future Plans

Proof of Concept

- We started with the assertion that it would be reasonable to add GDOI to an IKE implementation.
- A proof of concept test was done using the freely available *isakmpd* package to test the assertion.
- Result:It was easy to add with few changes to the original code

Current status

- Latest draft:

  http://www.ietf.org/internet-drafts/draft-ietf-msec-gdoi-00.txt
- Two independent implementations underway (Cisco and Nortel). Interoperability testing not yet begun.
- Group DOI protocol specification under evaluation by Catherine Meadows, NRL.

Group DOI

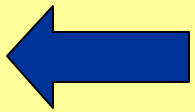
GDOI Design Criteria

Overview of related protocols

GDOI Protocol Overview

Current Status

Future Plans



Future Plans

- Revise the draft as necessary.
  - Fix inconsistencies
  - Investigate features necessary to support large groups.
  - Investigate using a different port number than IKE
- Complete the reference implementations and test with multicast applications.