RFC 3706

A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers

(Czerny Andeas)

Summery

1. Introduction
2. Keepalives and Heartbeats
3. DPD Protocol
4. Resistance to Replay Attack and False Proof of Liveliness

Situation

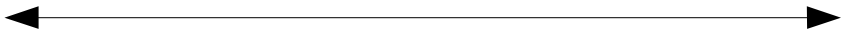
Peer A

Peer B

Internet

IKE

Protected IP traffic



IKE

Problem if connectivity goes down

- No way for IKE and IPSec to identify the loss of peer connectivity
- The SAs can remain until their lifetimes naturally expire

SA = Security Association

"Black Hole" Situation

- Packets are tunneled to oblivion
- It is often desirable to recognize black holes as soon as possible
  - failover to a different peer quickly.
  - recover lost resources.
- Proposals
  - sending periodic HELLO/ACK messages to prove liveliness.

Keepalives and Heartbeats

- Bidirectional "keepalive" message exchange

  - a HELLO followed by an ACK

  only on side is interested in liveliness

- Unidirectional "heartbeat" message exchange

  - a HELLO only

  both sides have to demonstrate liveliness

Scenario 1

Peer A

Peer B

A's 10 sec. timer elapses first

Sends HELLO to B

HELLO

Receives HELLO

Acknowledges A's liveliness

Resets keepalive timer

ACK

Sends ACK

Receives ACK as proof of B's liveliness

Reset keepalive timer

Scenario 2

Peer A

Peer B (dead)

A's 10 sec. timer elapses first

Sends HELLO to B — HELLO → (dead)

Retransmission timer expires

Message could have lost in transit

A increments error counter

Sends another HELLO → (dead)

....

Scenario 3

Peer A

Peer B

A's 10 sec. timer elapses first

Sends HELLO to B

HELLO

Receives HELLO as proof of A's liveiness

B's 10 sec. timer elapses

Receives HELLO as proof of B's liveliness

HELLO

Sends HELLO

Scenario 4

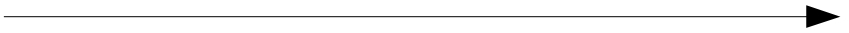
Peer A

Peer B

A's 10 sec. timer elapses first

Sends HELLO to B

HELLO



(dead)

....

Assumes B is dead

DPD Protocol

- A peer is free to request proof of liveliness when it needs it.
- asynchronous property allows fewer messages to be sent
- IPSec traffic itself serves as the proof of liveliness.
- Knowledge of the peer's liveliness, is only necessary if there is traffic to be sent.

DPD implementation

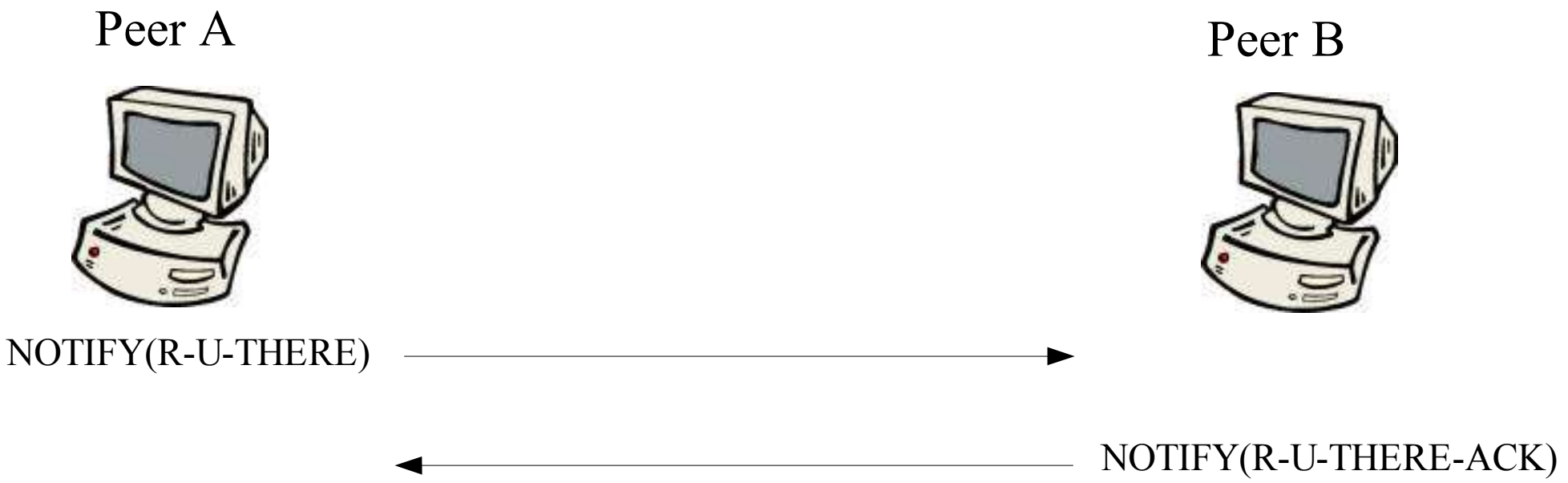
- In DPD, each peer can define its own "worry metric".
- the decision about when to initiate a DPD exchange is implementation specific.
- Each peer's DPD state is largely independent of the other's.

DPD Vendor ID

- Both peers of an IKE session MUST send the DPD vendor ID before DPD exchanges can begin

- MJR and MNR correspond to the current major and minor version of this protocol

Message Exchanges

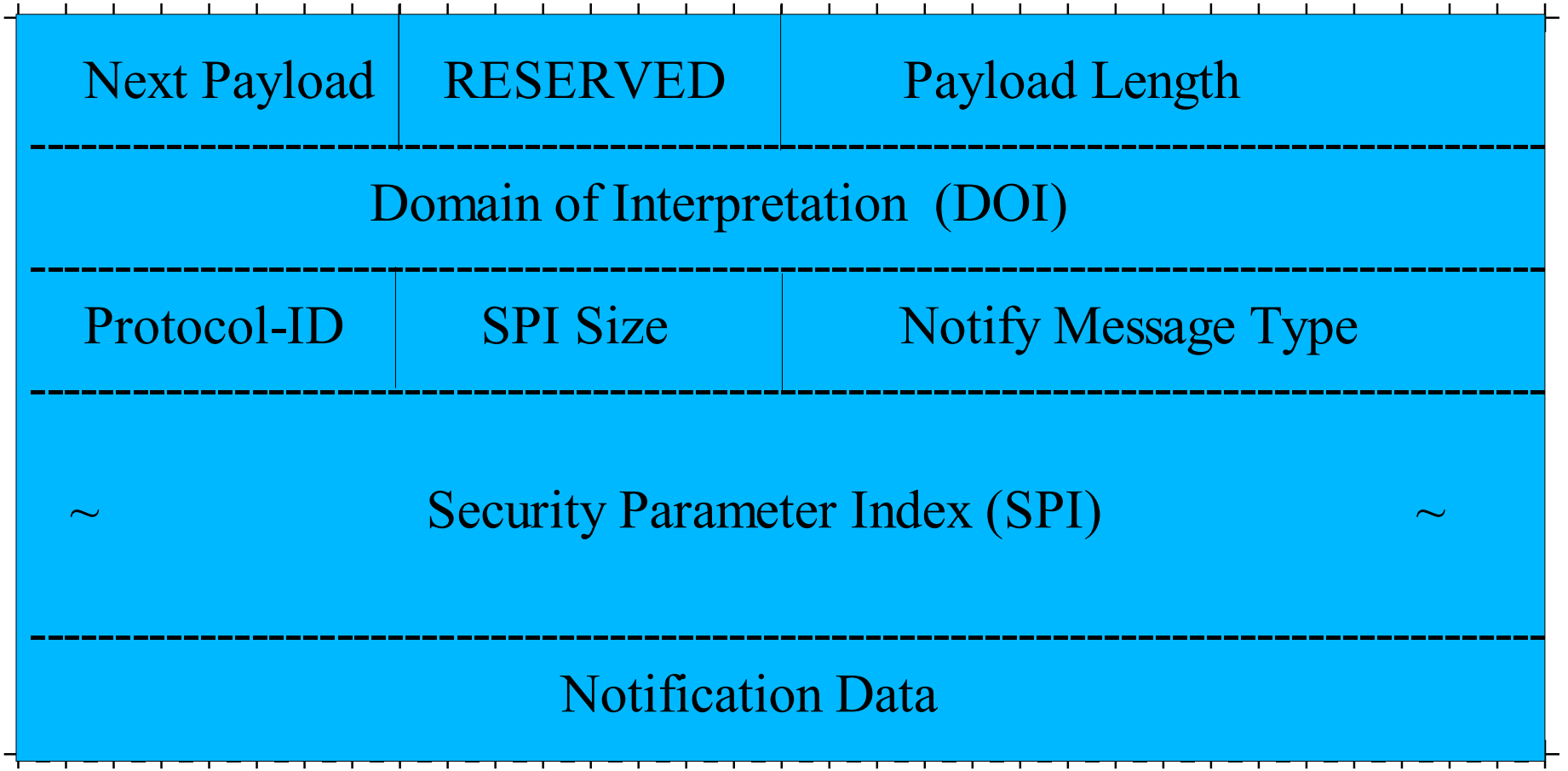


- The DPD exchange is a bidirectional message.
- Both messages are simply ISAKMP Notify payloads
  (Internet Security Association and Key Management Protocol)

Notify	Message Value
R-U-THERE	36136
R-U-THERE-ACK	36137

- A peer MUST keep track of the state of a given DPD exchange.
- Retransmit R-U-THERE queries when it fails to receive an ACK.
- After some number of retransmitted messages delete IPSec and IKE SAs to the peer.

Message Format



- Notify Message Type (2 octets) - MUST be set to R-U-THERE
- Security Parameter Index (16 octets) - SHOULD be set to the cookies of the Initiator and Responder of the IKE SA
- Notification Data (4 octets) - MUST be set to the sequence number corresponding to this message

Implementation Suggestion

- Liveliness of a peer is only questionable when no traffic is exchanged
- A viable implementation might begin by monitoring idleness.
- A peer's liveliness is only important when there is outbound traffic to be sent.
- initiate a DPD exchange if outbound IPSec traffic was sent, but not received any inbound IPSec packets.
- A complete DPD exchange will serve as proof of liveliness until the next idle period.

Comparisons

DPD vs keepalive/heartbeats

- Performance benefit:

  DPD do not need to sent regular messages.

  The number of IKE messages to be sent and processed is reduced.

- Implementation benefit:

  DPD needs only 1 timer

Resistance to Replay Attack and False Proof of Liveliness

Sequence Number in DPD Messages

- A responder to an R-U-THERE message MUST send an R-U-THERE-ACK with the same sequence number.
- The initial sender SHOULD reject the R-U-THERE-ACK if the sequence number fails to match the one sent with the R-U-THERE message.
- both SHOULD check the validity of the Initiator and Responder cookies presented in the SPI field of the payload.

Selection and Maintenance of Sequence Numbers

- both DPD peers can initiate a DPD exchange
- each peer MUST maintain its own sequence number
- The first R-U-THERE message sent in a session MUST be a randomly chosen number
- the high-bit of the sequence number initially SHOULD be set to zero.
- Sequence numbers MAY reset at the expiry of the IKE SA
- Maintain a window of acceptable sequence numbers

Benefit of sequence numbers

- detecting replayed messages

  prevents from needing to build, encrypt, and send ACKs.

- sequence numbers is that it adds an extra assurance of the peer's liveliness.