The SA Creation protocol in 06-449r5

T10/07-226r1
The Basics

☆ Three steps
  ✤ Get Capabilities (boring)
  ✤ Key Exchange
  ✤ Authentication

☆ Steps are identified in CDB
  ✔ SECURITY PROTOCOL field
  ✔ SECURITY PROTOCOL SPECIFIC field
The Basics
(continued)

☆ Four commands (in Key Ex. & Auth.)

+ SECURITY PROTOCOL OUT   Key
+ SECURITY PROTOCOL IN      Key
+ SECURITY PROTOCOL OUT     Auth
+ SECURITY PROTOCOL IN      Auth

☆ Always start at:

+ SECURITY PROTOCOL OUT   Key

☆ End after:

+ SECURITY PROTOCOL IN    Key
+ SECURITY PROTOCOL IN    Auth
The Basics
(Ladder Diagram)

Application
Client

SA Creation Active

SECURITY PROTOCOL
OUT Key Exchange

SECURITY PROTOCOL
IN Key Exchange

Device
Server

Think

Think

Think

May stop here.

Think

Think

SA Creation Active

SECURITY PROTOCOL
OUT Authentication

SECURITY PROTOCOL
IN Authentication

Application
Client

Device
Server

Think

Think

Think

May stop here.
Limit: 1 SA Creation per I_T_L Nexus
(Reflect this in the command protocol)

★ Lock-step protocol
  - Initiator waits for status on command \( n \) before sending \( n+1 \)
  - Target processes each command all the way to completion before sending status

★ Use REQUEST SENSE for progress indication
Limit: 1 SA Creation per I_T_L Nexus

★ Check multiple attempts only on SECURITY PROTOCOL OUT Key

★ Other Kindness Features
  ✤ Allow repeats on any SECURITY PROTOCOL IN
  ✤ SECURITY PROTOCOL OUT Auth returns a 'retryable' error if decryption/integrity check fails

✗ All other errors abort SA creation
Abandoning SA Creation

Initiator needs a way to tell target it does not want to continue SA Creation

✗ Delete created SA

✗ Delete *Creation Active* SA

✚ Valid in place of SECURITY PROTOCOL OUT Auth

✚ Requires minor changes to r4
Delete rules
SA Type/Usage too Public
(Hugo Krawczyk issue #1)

It is not hard to concoct scenarios where the SA Type or Usage information can help an attacker. For example, the attacker may learn, from the Usage information, which of the connections is for higher-security communications (say, labeled "top secret") and hence use this information to concentrate an attack or disrupt the connection, etc. It could also help in some forms of traffic analysis. Whether such scenarios are of concern for SCSI I cannot say.

✔ These SAs do not associate easily to connections
✔ Encrypting sensitive usage info using the SA is a way to hide it

★ No Changes Needed
I would change the name of the flag AUTH_NONE to signal the use of a combined encryption/integrity to AUTH_COMBINED. The existence of a AUTH_NONE flag may lead people to think that they can skip authentication all together.

☆ Outstanding idea

☆ Do it & assign it a non-zero codepoint to avoid IKEv2 entanglements