The SA Creation protocol in 06-449r5

T10/07-226r0
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
Why Worry?

IKE (and 06-449r4) manage protocol steps based on messages transferred (i.e., parameter data content)

06-449r5 manages protocol steps based on commands

→ Error recovery somewhat TBD

✓ Sanity check this change in direction
The Basics
(Ladder Diagram)

Application
Client

SA Creation Active

SECURITY PROTOCOL
OUT Key Exchange

SECURITY PROTOCOL
IN Key Exchange

Device
Server

Think

Think

Think

SA Creation Active

SECURITY PROTOCOL
OUT Authentication

SECURITY PROTOCOL
OUT Authentication

May stop here.
Limit: 1 SA Creation per I_T_L Nexus

⭐ Needs a special
  'I'm doing another one’ ASC

... but ...

✔ Where to check this?

📖 How much can a man-in-the-middle
do without giving himself away?
Limit: 1 SA Creation per I_T_L Nexus

(Choices)

✔ SECURITY PROTOCOL OUT Key
  ✶ Normal (non-sick initiator) beginning (r5)
  ✶ Other *bogus* SECURITY PROTOCOL commands break SA Creation

✔ Any SECURITY PROTOCOL OUT/IN
  ✶ More lenient (r4)
  ✶ If SECURITY PROTOCOL IN repeats are free, this hides man-in-the-middle
Abandoning SA Creation

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

✗ Delete created SA (when target thinks SA is okay but initiator doesn't)

✗ Otherwise:
  ✗ Send bogus SECURITY PROTOCOL OUT command (r4 & r5)
  ✗ Send special SECURITY PROTOCOL OUT command (new TBD function)