KARP KMP-Using IKEv2 with TCP-AO

draft-chunduri-karp-using-ikev2-with-tcp-ao-00

Uma Chunduri, Albert Tian

Ericsson Inc.

Joe Touch

USC/ISI

IETF 82, Taipei, Taiwan

November 13-18, 2011

Using IKEv2 with TCP-AO

For TCP based routing protocols BGP [RFC4271], PCEP [RFC5440], MSDP [RFC3618] and LDP [RFC5036] – to move away from existing MD5 based manual mechanism:

- RFC 5925: TCP-AO (Key agility, Algo. agility, replay protection etc.)
- RFC 5926: TCP-AO algs. (specific algs. and parameters)

- RFC 5996: IKEv2 Key Management protocol (flexible and yet strong KMP)

Using IKEv2 with TCP-AO

The Problem:

How to integrate TCP based pair wise routing protocols (BGP, LDP, MSDP, PCEP) with Key Management Protocol (KMP)?

Using IKEv2 with TCP-AO

Goals:

Minimize changes to all TCP based Routing Protocols to integrate with KMP

- by Using TCP-AO's infrastructure (MKTs)

Extending IKEv2 to negotiate RP SAs

- to continuously benefit from new IKEv2 features
  E.g. Pre-shared key only and yet secure authentication

Using IKEv2 with TCP-AO (cont.)

1st Question on IKEv2: Which peer authentication is suitable for RPs ?

- Symmetric Shared key based
  - Pre-shared key only options worked out by ipsecme WG
- Asymmetric (PKI)
  - RSA, DSS
  - ECDSA
- EAP Based (EAP Only - RFC5998)
  - Non Client/Server mode
    - PAX (RFC 4746)
    - EAP-pwd (RFC 5931)
    - EKE based (RFC 6124)

Using IKEv2 with TCP-AO (cont.)

What is needed from IKEv2

- WG: One peer authentication mechanism suitable for RPs
- Extensions to Security Association (SA) Payload for tcp based routing protocol SA
    - extensions required listed in the draft (non IPSec DOI)
- Simplified Traffic Selectors

Using IKEv2 with TCP-AO (cont.)

BGP Multisession Requirement

- Multiple TCP sessions between same peers per AFI/SAFI
  - ietf-idr-bgp-multisession-06
- Each TCP session can have different rekey lifetime
- Each session can be differentiated by different SIP
  - Multi-session draft tries to avoid the same
- Sessions must be differentiated by their transport information
  - Currently done by different IP addresses => undesirable
  - Could be done by different dest ports (services) => undesirable
  - Could be done by different source ports => requires code
    - Implies a separate TCP-AO MKT for each session
    - Source port needs to be fixed by BGP or in a library before MKT can be negotiated

Using IKEv2 with TCP-AO (cont.)

Crypto Key Tables

- It's a database of all the keys and for all protocols (interfaces or more specific protocol info)
- It specifies the selection process (equivalent to Association lookup) once these are populated

Using IKEv2 with TCP-AO (cont.)

…..

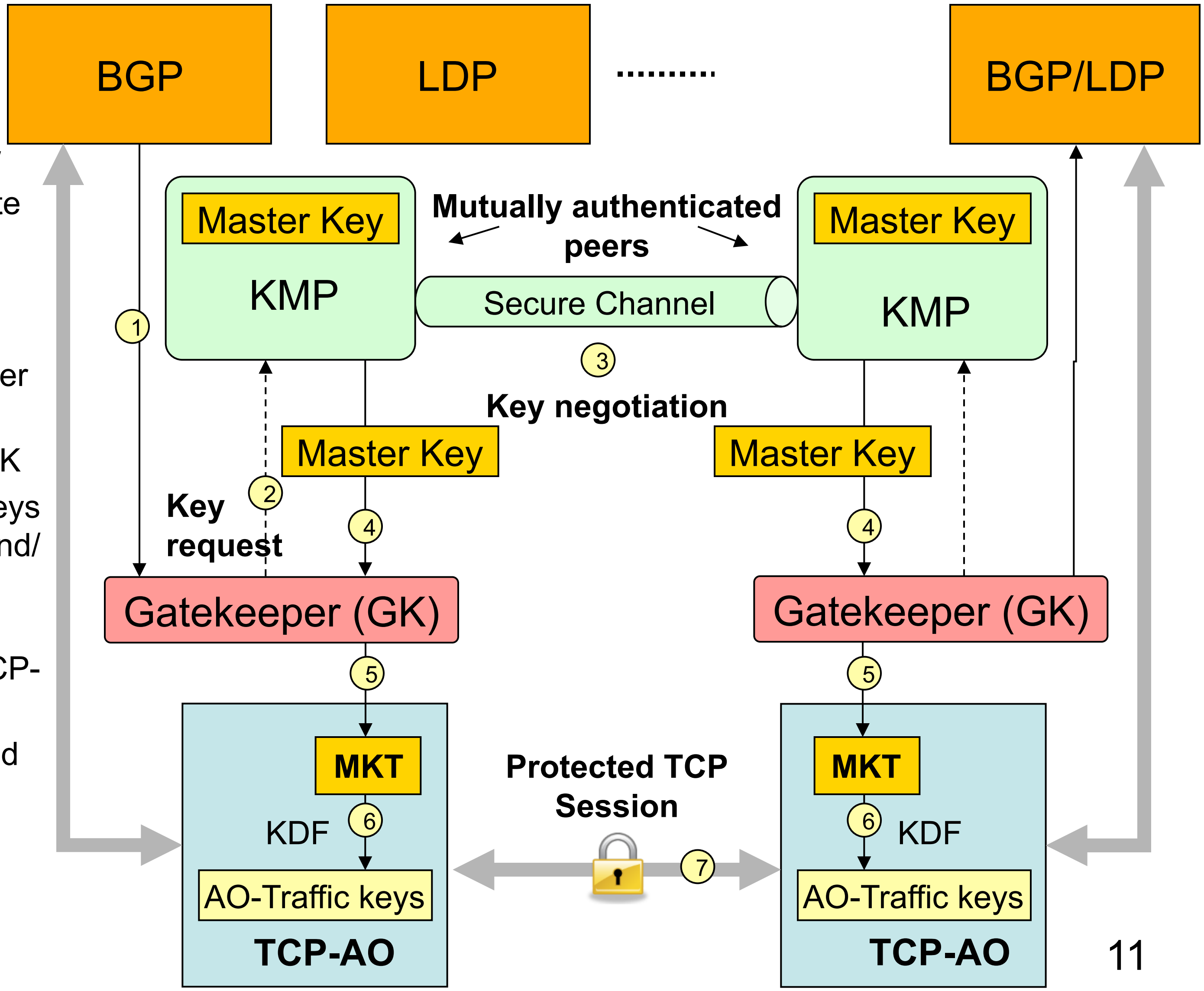
- All routing protocols need to trigger KMP to get the SA
- All routing protocols need to maintain the same with the lifetime
- and rekey when lifetime expires

(Essentially complete SA management at each RP level)

Using IKEv2 with TCP-AO (cont.)

Solution

1. BGP/LDP sets configured Auth/KDF/ lifetime info and initiate TCP connection
2. GK triggers KMP (IKEv2)
3. IKEv2 negotiate Master key
4. Master keys add to GK
5. GK converts IKEv2 keys into MKTs; revokes and/ or retriggers IKE as needed
6. Use KDF to derive TCP-AO traffic-keys
7. TCP session protected



Using IKEv2 with TCP-AO (cont.)

What is needed from TCP-AO

- Transport-level differentiation of multisession BGP sessions
  - Socket pair must be unique
  - Currently use different IP addresses
  - Use different source ports => need code somewhere (BGP source, link library, OS)
- IKEv2-compatible keying support
  - IKEv2 assumes IPsec manages SA timers, triggers new SA requests
  - TCP-AO assumes external key management, incl. timers and rekey initiation
  - Need separate key timers, rekey initiation ➔ Gatekeeper (GK) (see: *Ghostbusters*)
- Result
  - IKEv2 generates keys and parameters
  - GK triggers IKEv2 initial and rekeying, inserts info into TCP-AO, revokes keys
  - TCP-AO implements transport authentication based on given info.

Using IKEv2 with TCP-AO (cont.) Advantages

- No TCP based routing protocol changes
  - Transparent to keys and KMP
  - Configuration can be *similar* to manual keys with TCP-AO
- No Extensions for TCP-AO (5925)
- Minimal Extensions for IKEv2 (5996) to negotiate non-IPSec SA for RPs
  - Simplified configuration for RPs
- Gatekeeper isolates how TCP-AO mimics IPsec to IKEv2
  - Manages the state/timers that IKEv2 expects IPsec to manage
- Leaves BGP source port lockdown as implementation issue
  - Many solutions, including rewrite BGP, relink to a shim library, revise OS
  - E.g., convert *connect(srcIP, *, dstIP, bgp-port, USE_AO) to*
    - *bind(srcIP, *, dstIP, bgp-port) => source port selected at bind time*
    - *getsockname(…) => returns source port*
    - *setsockopt(TCPAO, full socket info, keys, etc.) => set MKT based on full socket pair*
    - *connect(as usual) => finish connect*

Questions & Comments?

Thank You!