A Multicast Key Management Protocol (MKMP)

draft-ietf-<somewhere>-mkmp-00.txt

Dan Harkins
cisco Systems
dharkins@cisco.com

Naganand Doraswamy
Bay Networks
naganand@baynetworks.com

MKMP Features

- ***Scalable, secure (of course) distribution of multicast keys***
  - -- access control is enforced
  - -- key distribution is mutually authenticated
  - -- key distribution provides live-ness proof
- ***Does not unnecessarily burden intermidiate communications devices (routers need not join the secure multicast group)***
- ***Independent of underlying multicast routing protocol***
- ***Uses IPSec to secure multicast traffic; uses ISAKMP/Oakley-type messages for key management***
- ***Allows for incremental deployment***

10000m view of MKMP

- ***MKMP-aware routers can become Group Key Distributors to allow key acquistion to scale***
- ***A Group Information Tuple, created by the group key creator, enforces access and delegates key distribution authority to Group Key Distributors***
- ***Uses a 224.0.1.x group-- ALL-MKMP-DEVICES***

  -- all MKMP-aware entities on the network listen to this group

- ***Key Acquire is separate from group join***

Creation of a Secure Group

- ***Group Key Manager creates a key, a list of candidate key distributors, and access control information***

- ***Announcement of secure group done via SAP/SDP***

  -- SDP description for Group Key Manager

  k=uri:mkmp://bad.attitu.de
  k=addr:IN IP4 199.166.37.16

Key Distribution Delegation

- ***Periodic Group Key Distributor solicitation sent to the multicast group address. Message contains a list of all candidate distributors and corresponding "cookies".***

  -- cookie state need not be retained as long as bogus cookies can be rapidly identified (Karn and Simpson)

- ***Message sent with Router Alert Option (RFC 2113)***

  -- all routers on the distribution tree will examine the message

  -- MKMP-aware routers can parse; non-MKMP-aware routers merely forward the message, albeit on the slow path

- ***If the message reaches a candiate Group Key Distributor it obtains the key from the Group Key Manager using the Key Distribution Protocol***

  -- GIT identifying router as Group Key Distributor and containing access control information is also obtained

Key Distribution Delegation (cont.)

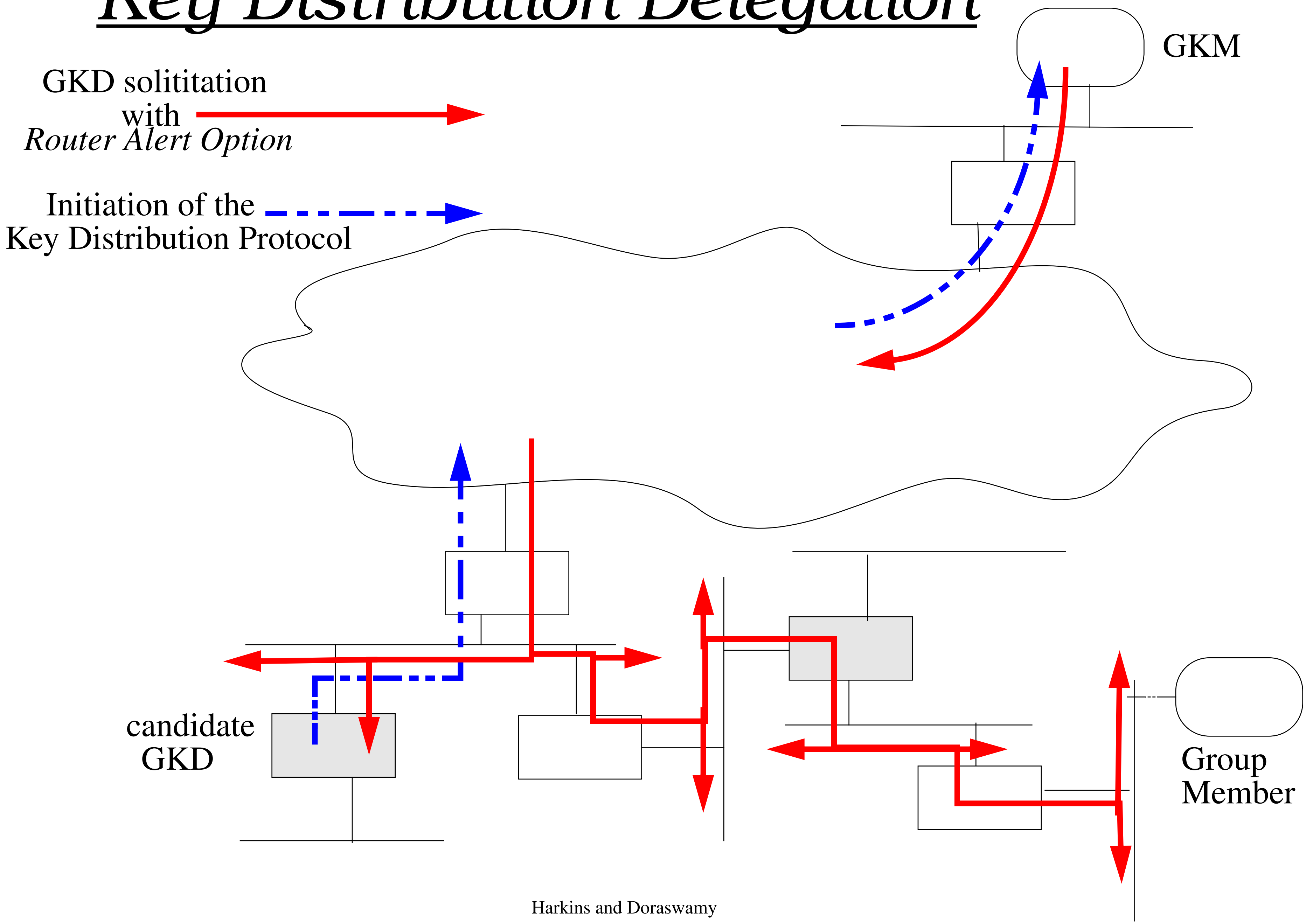
- ***Only routers already on the distribution tree become GKDs.***

  -- not all bad since GKDs will only appear in places they're needed.

- ***GKDs must prevent themselves from being pruned***

  -- e.g. send periodic join/prune messages even if there are no longer any members downstream

Key Distribution Delegation



Group Key Acquisition

- ***Host wishing to obtain Group Key probes its neighborhood for Group Key Distributors***

  -- message sent to the ALL-MKMP-BOXES group
  -- scoped by TTL
  -- original TTL of probe included in message

- ***Group Key Distributors set a response timer based upon distance from candidate member***

- ***Group Key Distributor responds with an assert***

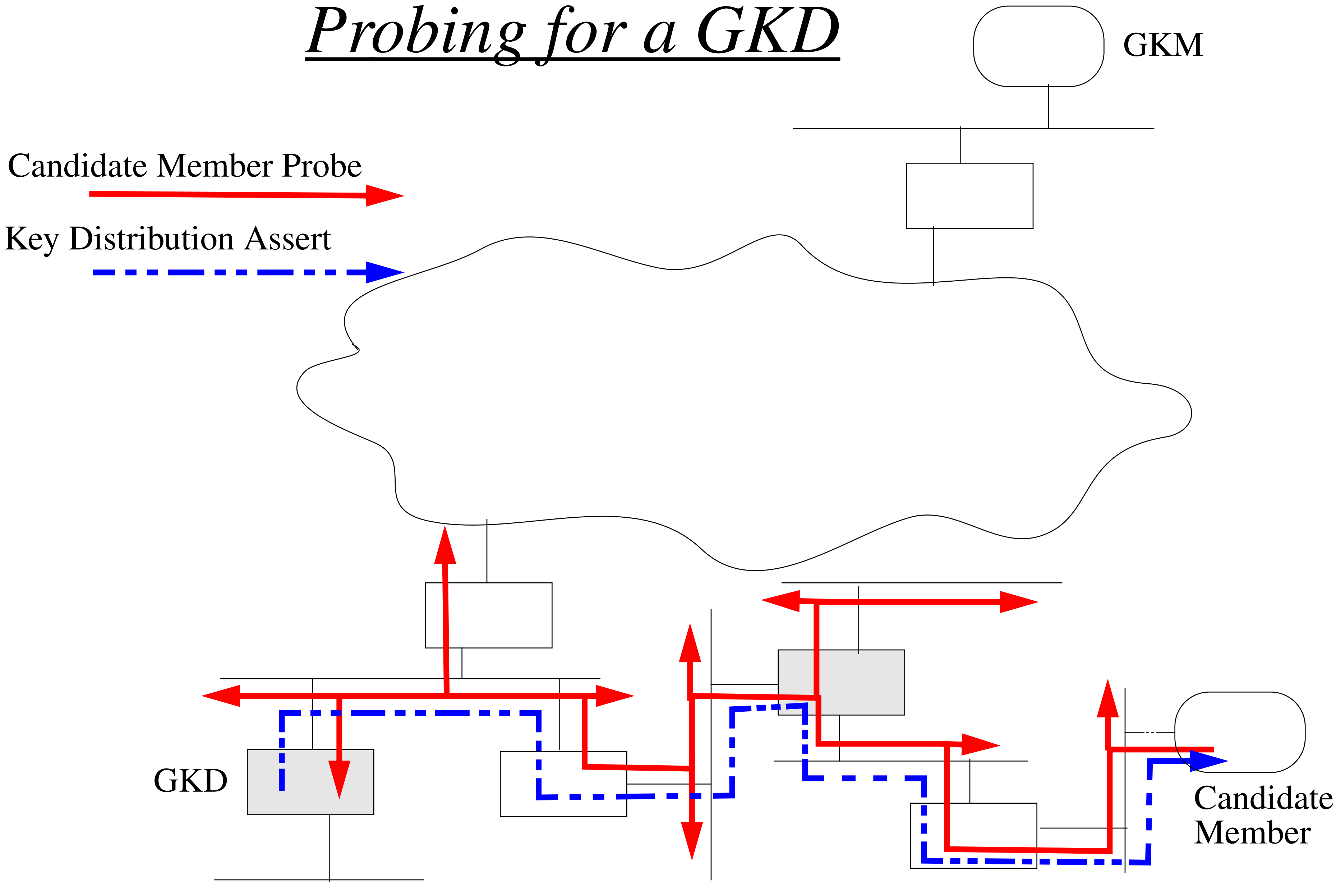
  -- GKD's GIT, signed by the GKM, asserts distribution authority
  -- assert also contains a "cookie"

- ***Candidate member squelches further responses with a "got it" message multicast to ALL-MKMP-DEVICES***

- ***If there are no responses host contacts Group Key Manager Directly and obtains the key using the Key Distribution Protocol***

  -- initially quite frequent but as hosts join the group and a multicast distribution tree is built, becomes less frequent

Probing for a GKD



Key Distribution Protocol

Candidate Member | **Key Holder (GKM or GKD)**

hdr, IDcm, { N_1 } $_{pub\_kh}$, [CERT,] HASH1 →

← ***hdr, { N_2 } $_{pub\_cm}$, SAm, SAc, [GIT,] HASH2***

N_1 = random number of Candidate Member's choosing

$N_2 = N_1$ ^ *key*

HASH1 **= SHA (cookie-R | cookie-I | N_1 | IDcm_b)**

HASH2 **= SHA (cookie-I | cookie-R | N_2 | N_1)**

cookie-I = cookie from GKD assert or GKM solicitation

cookie-R = random cookie of Candidate Member's choosing

IDcm = identity of the Candiate Member

Key Distribution Protocol (cont.)

- ***SAm: attributes + SPI used to protect multicast traffic***

  -- GKDs need not retain this Security Association since they do not join the multicast group

- ***SAc: attributes + SPI used to send MKMP control messages to all members***

  -- group rekey: new security associations and new key sent under the protection of the old SAc

  hdr, SAm, SAc, KE, SIG

  with the Router Alert Option set to ensure all GKDs obtain message.

  -- group termination

  -- notification of forced member removal and need for global key reacquire-- new SAm with SPI of zero.

Rekey After Forced Member Removal

Group Member | **Key Holder (GKM or GKD)**

hdr, IDcm, HASH1 $\longrightarrow$

hdr, { N_2 } $_{pub\_cm}$, SAm, SAc, [GIT,] HASH2 $\longleftarrow$

HASH1 **= SHA (cookie-R | cookie-I | $N_{1\text{-orig}}$ | key)**

HASH2 **= SHA (cookie-I | cookie-R | N_2 | $N_{1\text{-orig}}$)**

$N_{1\text{-orig}}$ = original Candidate Member nonce (N_1)

key = old key

N_2 = $N_{1\text{-orig}}$ ^ new-key

IDcm = identity of group member

What Next?

- ***Clean up draft-ietf-<somewhere>-mkmp-00.txt and find a home-- ipsecond? multicastsec?***
- ***Ask IANA for a 244.0.1.x group for ALL-MKMP-DEVICES***
- ***Reference Implementations?***