MLS for IPsec Group Key Management

draft-kohbrok-ipsecme-mls-gike

Konrad Kohbrok
Raphael Robert
Valery Smyslov

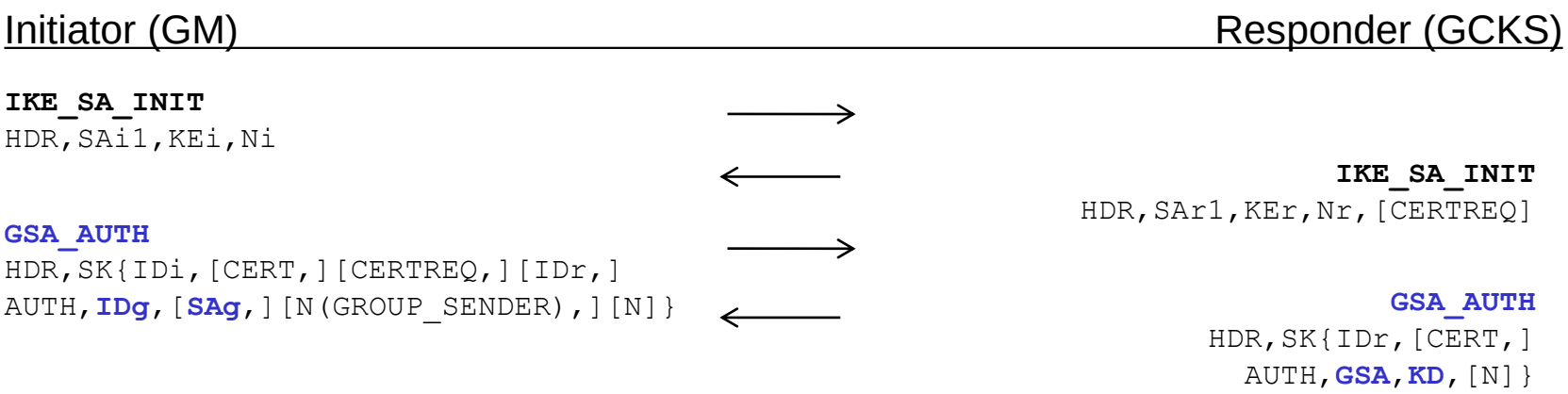
IPsec Workshop, Vienna, July 2026

Securing IP Multicast

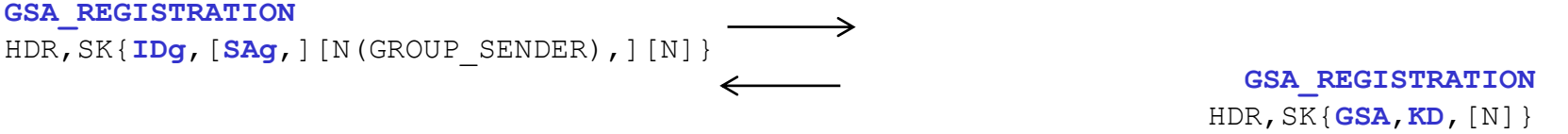
- IP multicast applications
  - Contain at least 1 sender, and N receivers
  - Take advantage of the network to route and replicate IP packets, such that the same packet reaches all N receivers
- This requires senders and receivers to share setup an IPsec SA using the same keys
  - The IPsec policy and keys are not individually negotiated, but instead of distributed by a Group Controller / Key Server (GCKS) to Group Members (GMs)
  - A GM invokes a unicast Registration protocol to authenticate to the GCKS. The GCKS then authorizes the GM, and distributes IPsec policy and keys to the GM.
  - A Rekey protocol enforces a time-based key rollover strategy
- This architecture is specified in RFC9838 (G-IKEv2)

G-IKEv2 Registration

- Initial registration (no IKE SA between GM and GCKS)

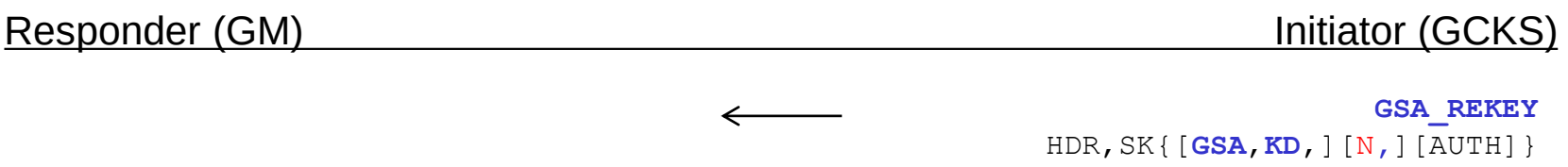


- Subsequent registration (IKE SA has already been created)

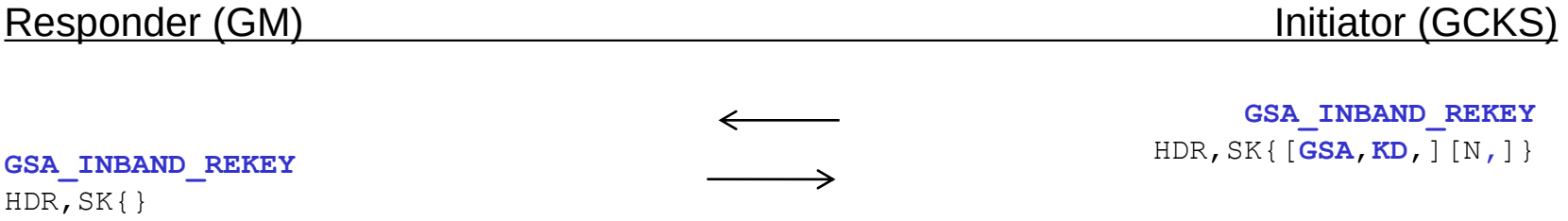


G-IKEv2 Rekey

- Multicast rekey: intended for large groups, protected by policy previously distributed by the GCKS



- Unicast rekey: intended for small groups, used registration IKE SAs with each GM



G-IKEv2 Architecture

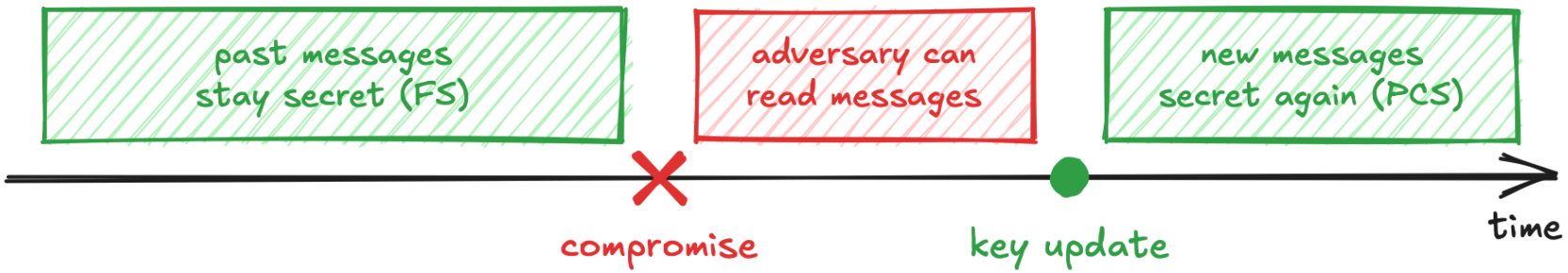
- GCKS authenticates GMs and authorizes them to the group
- GCKS controls all group properties, including traffic selectors
- GCKS generates all group keys, including IPsec keys
  - even when GCKS is not a member of the group, it can read its traffic
- GMs play passive role in key management – they do not contribute into group keys

Introduction to MLS

- RFCs 9420 (spec) and 9750 (architecture)
- Secure messaging and group key agreement protocol
- Efficient (sublinear) key updates, even in large groups
- PQ-ready through KEM and signature abstraction
- Group members agree on group state cryptographically
  - Group membership
  - Other, application-defined state (e.g. policy)

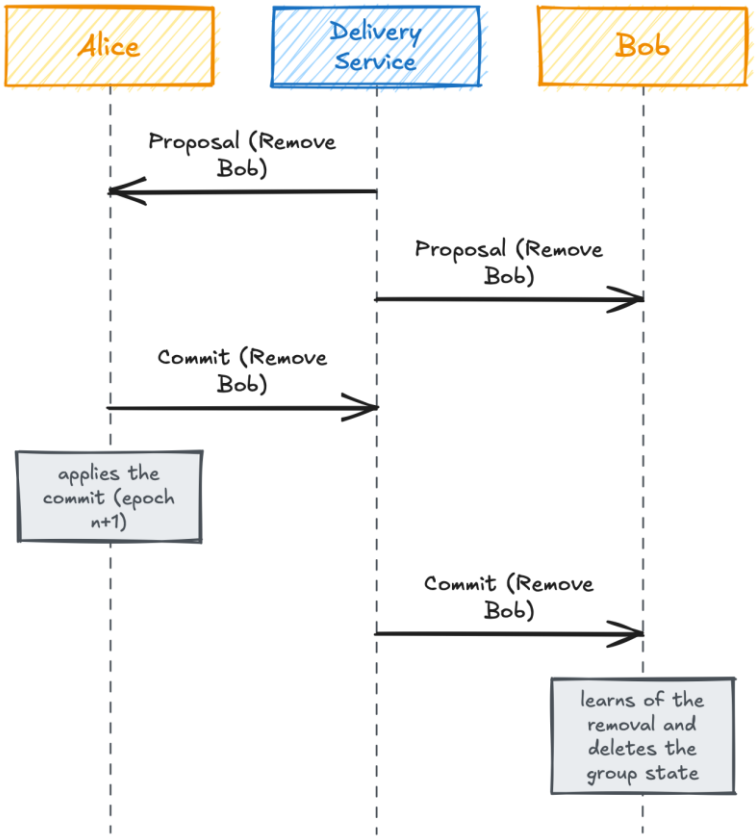
Introduction to MLS

- Key updates provide forward secrecy (FS) and post-compromise security (PCS)



Introduction to MLS

- Group members or external senders propose group action(s), one group member commits



Related Works

- The CURRENT BoF at IETF 126
- Drafts
  - Securing QUIC with MLS (draft-tian-quic-quicmls)
  - The MLS-TLS secure channel protocol (draft-kohbrok-mls-tls)

Use of MLS in G-IKEv2

- GCKS acts as MLS Authentication Server (AS), Distribution Server (DS) and external sender
  - MLS objects are delivered to GM using either unicast IKE SAs or multicast Rekey SA (managed by GCKS)
- **IPsec keys are generated using MLS**
  - MLS exporters are used to compute the keys
- MLS Profile
  - external_senders extension must be supported
    - lets the GCKS propose Add and Remove operations without making the GCKS an MLS member
  - all MLS Proposals and Commits are sent as MLS PublicMessages
    - lets the GCKS reconstruct the public tree from the handshake stream
  - joining through External Commit is not supported

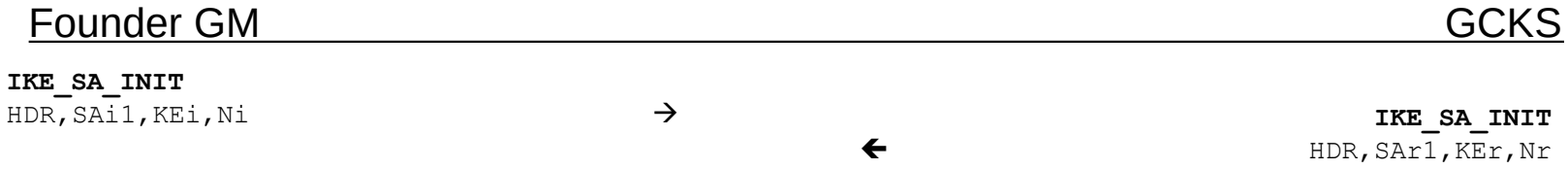
Changes to G-IKEv2

- All G-IKEv2 payloads and all unicast exchanges are re-used (with some restrictions on the payload content)
- Optional re-use of multicast `GSA_REKEY` pseudo-exchange for distribution of MLS objects
  - LKH key management can be re-used to control GM membership in the Rekey SA group
- New exchanges
  - `GSA_MLS_REQUEST` – GCKS requests MLS commit object from designated GM (chosen on its own)
  - `GSA_MLS_UPLOAD` – GM sends MLS object(s) to GCKS
- No new payloads
- New notifications
  - `MLS_OBJECT` (status) – contains MLS object(s)
  - `MLS_COMMIT_REJECTED` (error) – GCKS rejects commit

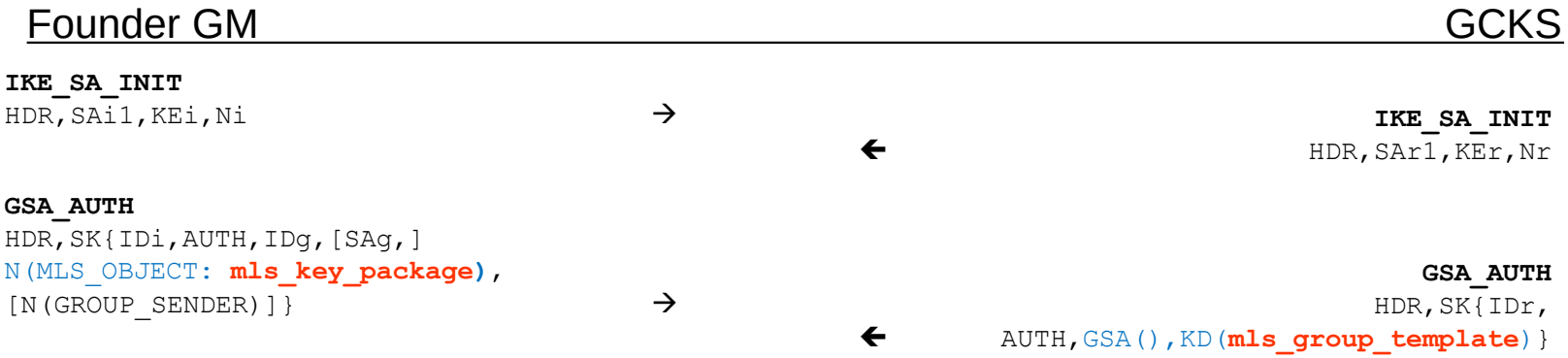
Operations of G-IKEv2 with MLS

- Following operations are defined
  - Group creation
  - Registration of a GM
  - Removing of a GM
  - GM-initiated key update

Group Creation



Group Creation



Group Creation

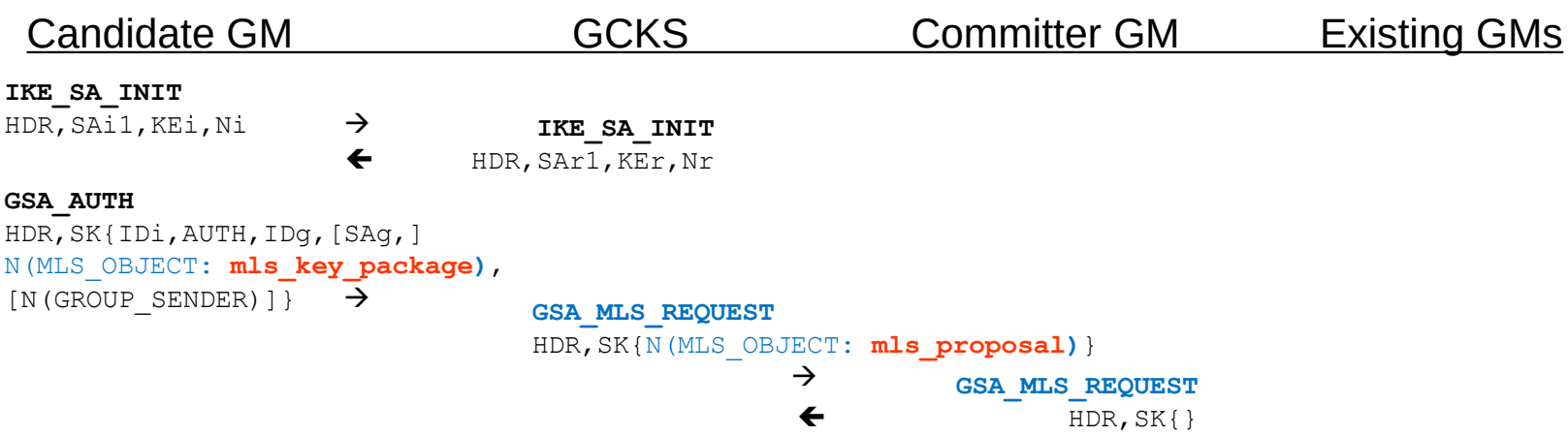
GM Registration

Candidate GM		GCKS	Committer GM	Existing GMs
IKE_SA_INIT				
HDR,SAi1,KEi,Ni	→	**IKE_SA_INIT**		
	←	HDR,SAr1,KEr,Nr		

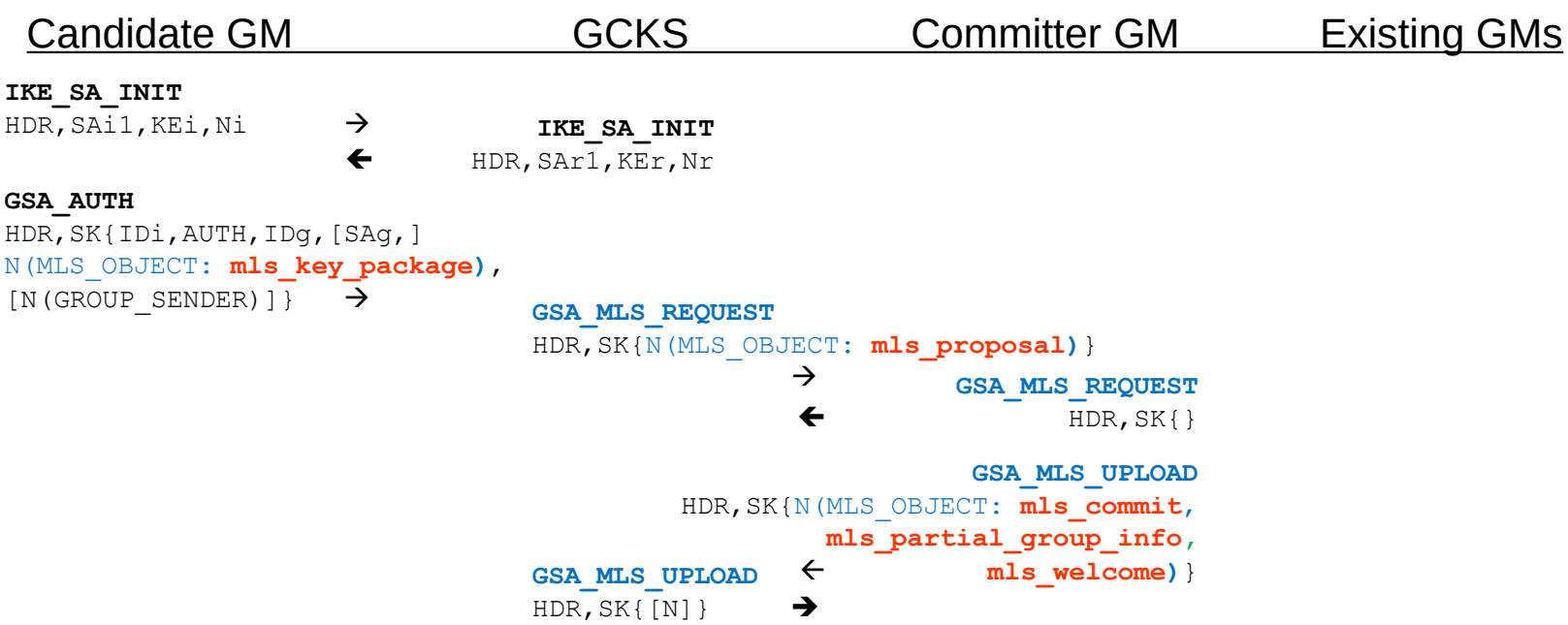
GM Registration

Candidate GM		GCKS		Committer GM	Existing GMs
IKE_SA_INIT					
HDR,SAi1,KEi,Ni	→		**IKE_SA_INIT**		
	←		HDR,SAr1,KEr,Nr		
GSA_AUTH					
HDR,SK{IDi,AUTH,IDg,[SAg,]					
N(MLS_OBJECT: **mls_key_package**),					
[N(GROUP_SENDER)]}	→				

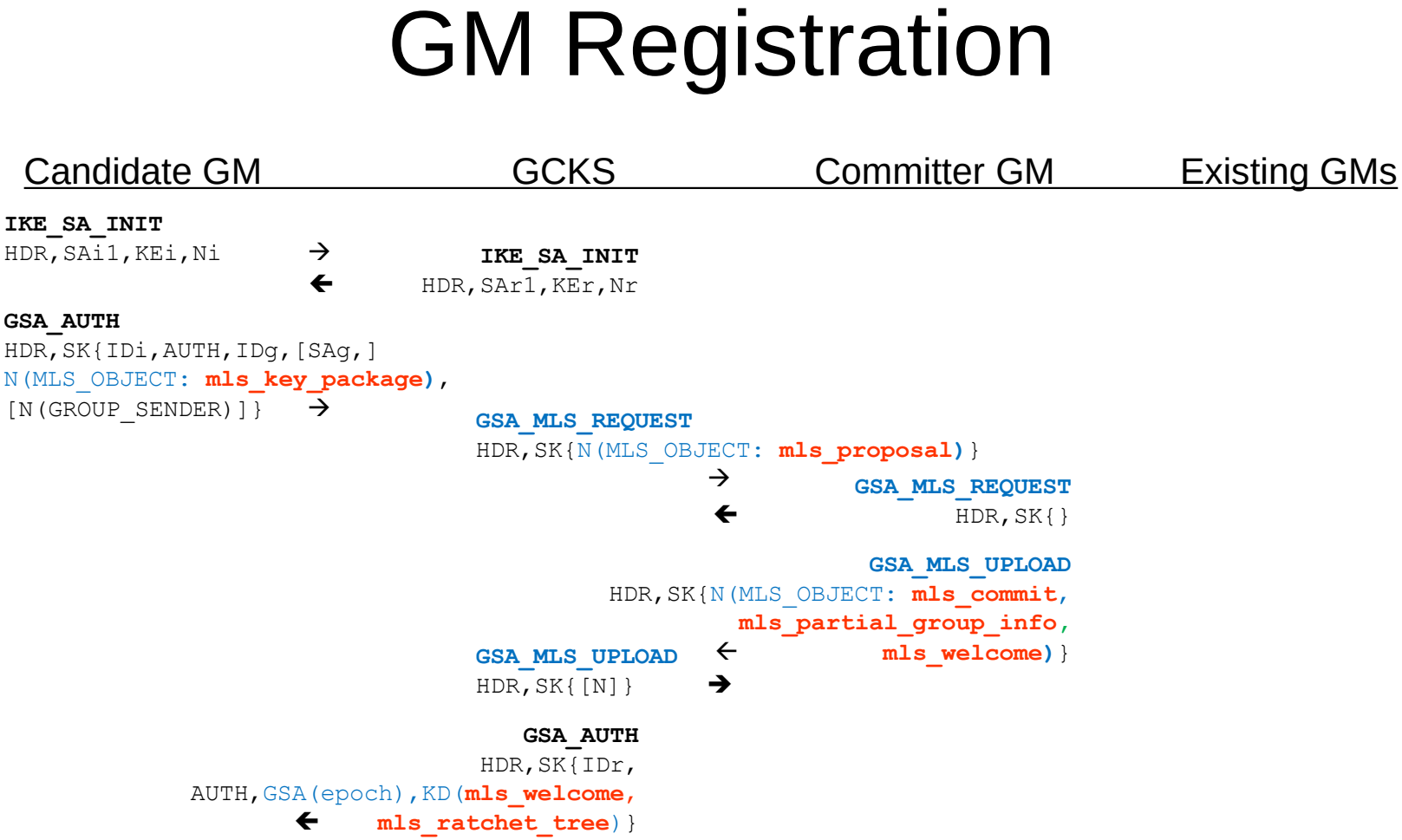
GM Registration



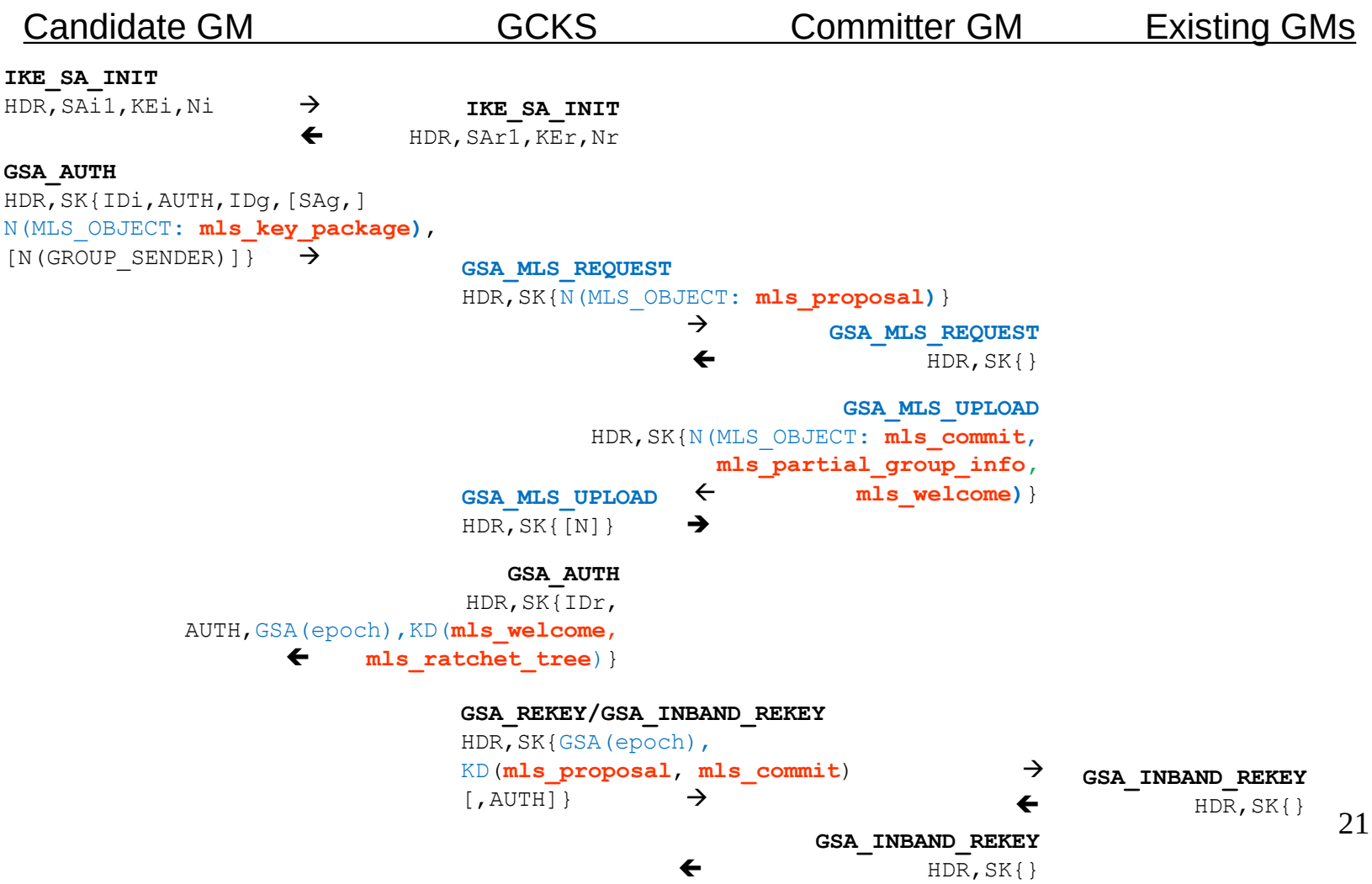
GM Registration



GM Registration

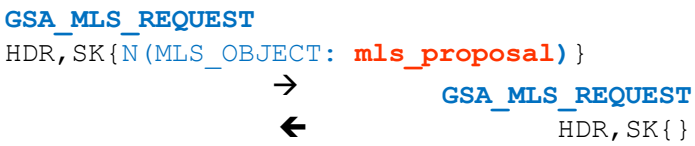


GM Registration

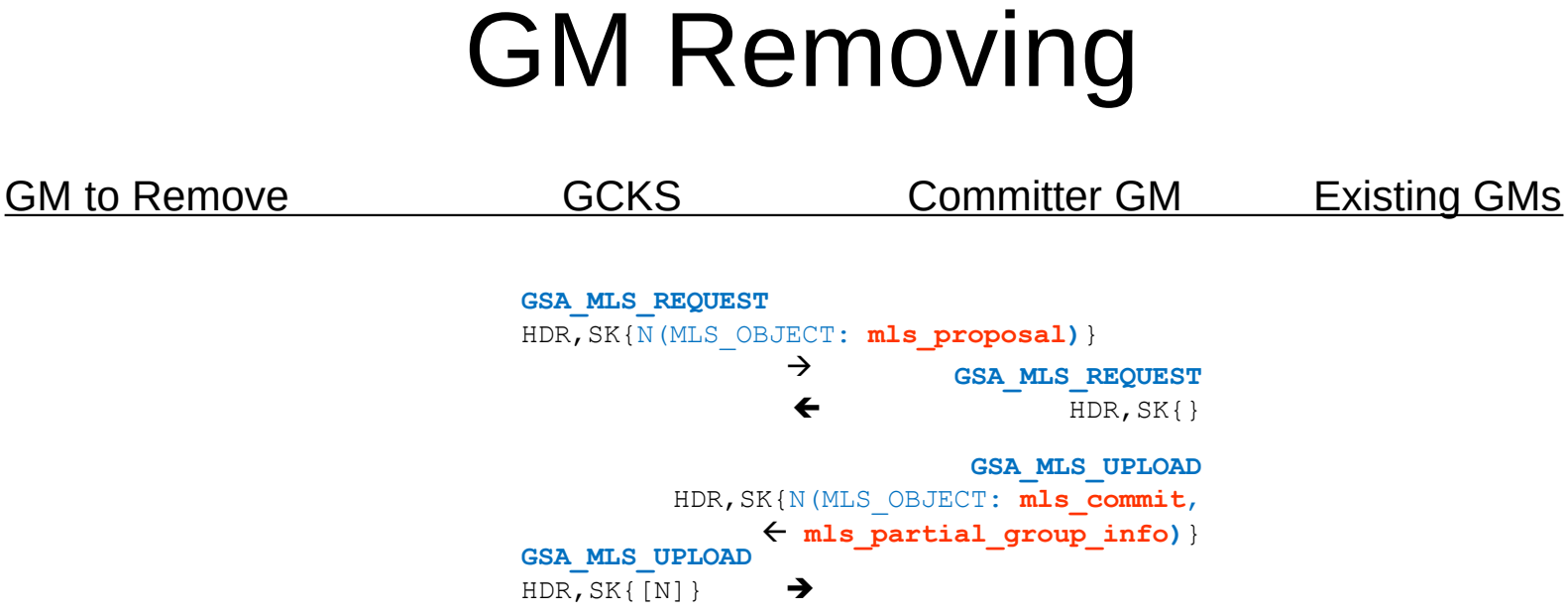


GM Removing

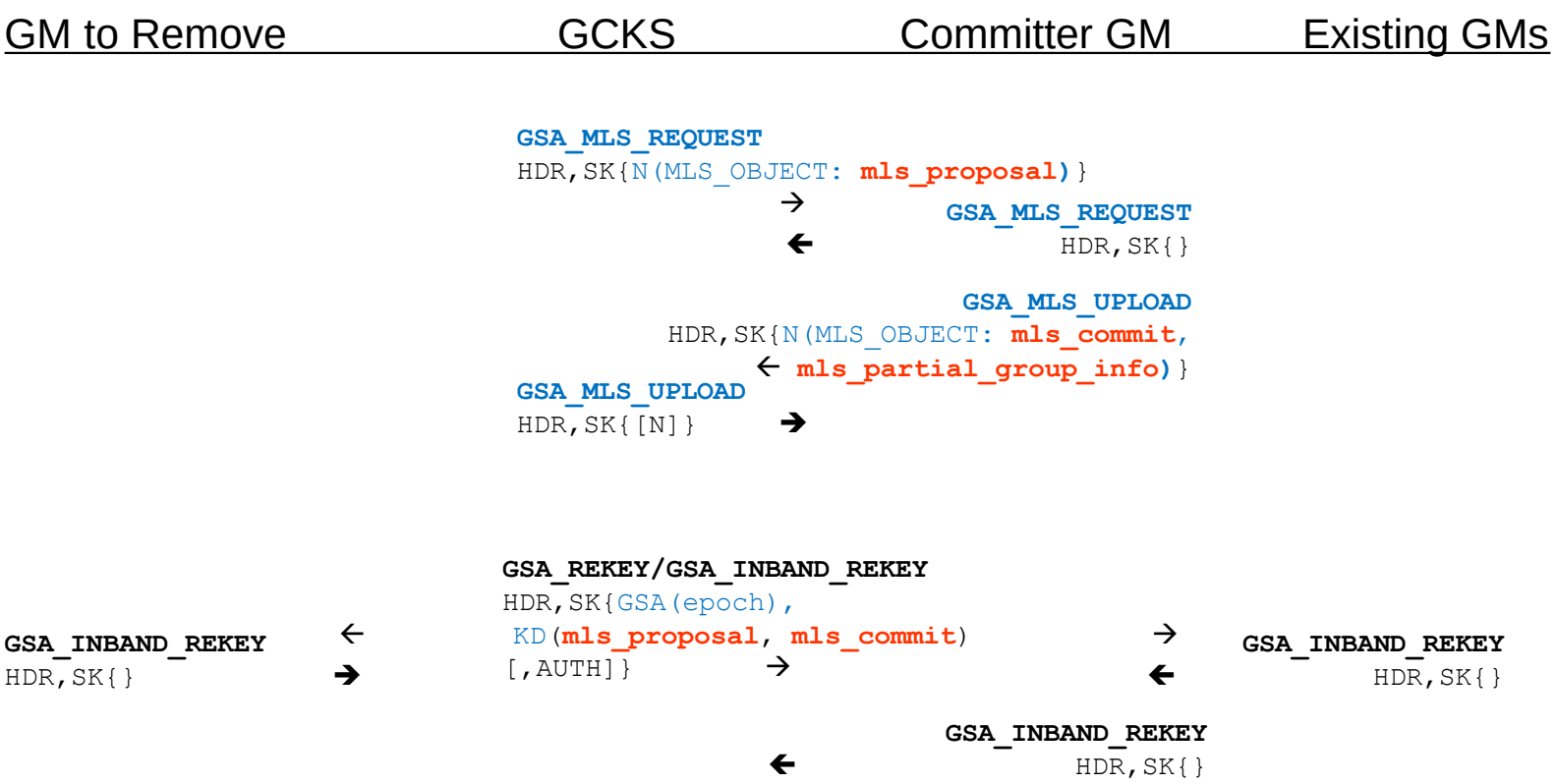
GM to Remove	GCKS	Committer GM	Existing GMs



GM Removing

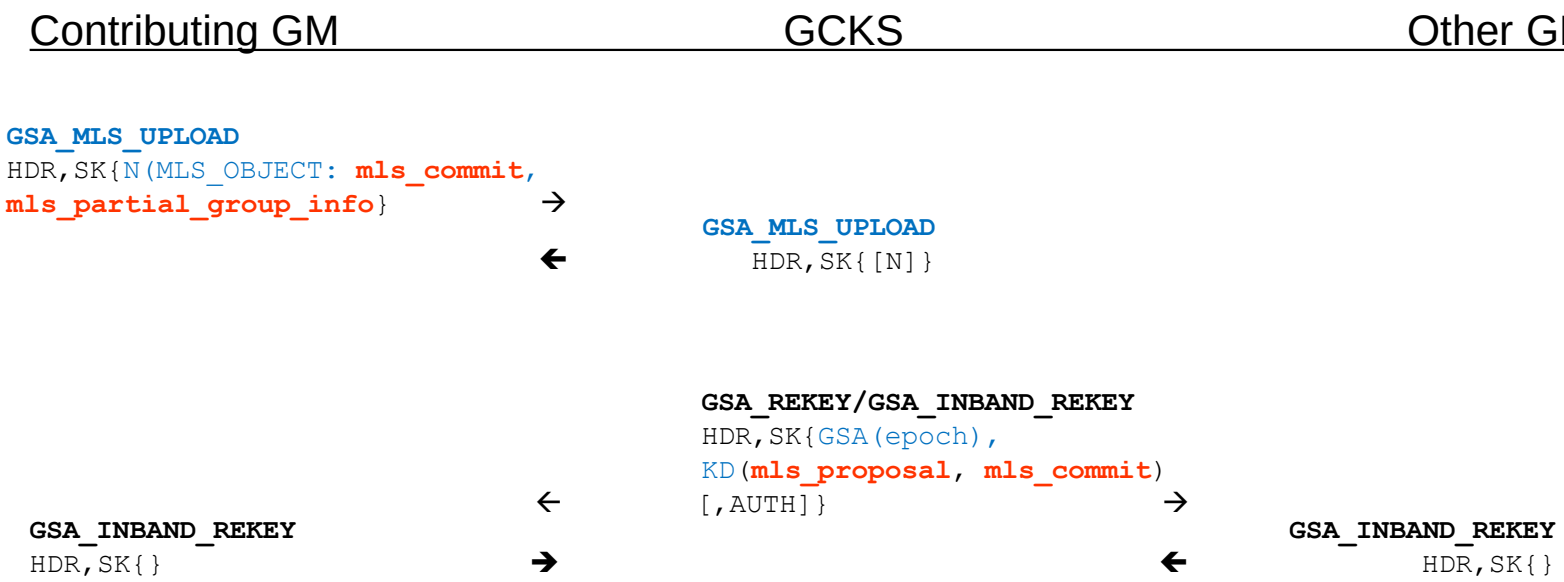

GM to Remove
GCKS
Committer GM
Existing GMs
GSA_MLS_REQUEST
HDR,SK{N(MLS_OBJECT: mls_proposal)}
GSA_MLS_REQUEST
HDR,SK{}
GSA_MLS_UPLOAD
HDR,SK{N(MLS_OBJECT: mls_commit,
mls_partial_group_info)}
GSA_MLS_UPLOAD
HDR,SK{[N]}

GM Removing


GM to Remove
GCKS
Committer GM
Existing GMs
GSA_MLS_REQUEST
HDR,SK{N(MLS_OBJECT: mls_proposal)}
GSA_MLS_REQUEST
HDR,SK{}
GSA_MLS_UPLOAD
HDR,SK{N(MLS_OBJECT: mls_commit,
mls_partial_group_info)}
GSA_MLS_UPLOAD
HDR,SK{[N]}
GSA_REKEY/GSA_INBAND_REKEY
HDR,SK{GSA(epoch),
KD(mls_proposal, mls_commit)
[,AUTH]}
GSA_INBAND_REKEY
HDR,SK{}
GSA_INBAND_REKEY
HDR,SK{}
GSA_INBAND_REKEY
HDR,SK{}

GM-initiated Key Update

GM-initiated Key Update



G-IKEv2 Architecture with MLS

- GCKS authenticates GMs and authorizes them to the group
- GCKS controls all group properties, including traffic selectors
- GCKS does not generates group IPsec keys, it still generates keys for multicast Rekey SA
  - when GCKS is not a member of the group, it cannot read its traffic
- GMs play active role in key management – they collectively generate IPsec group keys

Thanks!