LAN-based Key Server (LKS) for IEEE 802.1af

Brian Weis

Agenda

- Why LKS?
  - Or to put it another way, Why "Yet Another MACsec Key Agreement" Protocol?
- General MACsec Key Agreement (MKA) design goals
- Overview of LKS
  - Design Principles
  - State Machine
- Comparison to KSP
- Example LKS Message Flows

Why LKS?

- A reasonable question is "Why consider a new model when KSP has already been proposed?"
- In fact, I reviewed KSP quite thoroughly
  - Given the environmental constraints, the overall design is reasonable
    - IEEE 802.1AE needs a group security model without pair-wise security associations (unlike IEEE 802.11i)
    - But it is important to recognize that the only available security model is one of complete and equal trust of all group members!
  - Discussing KSP with Mick provided a lot of background and appreciation for its features
- Still, there are some uncomfortable bits that led to me consider an alternative

Uncomfortable bits

- KSP key management methodology is unconventional
  - Since KSP was first introduced, NIST has released key management guidelines (NIST SP 800-57 Parts 1 & 2).
    - The KSP key contribution method cannot be fully described to match those guidelines.
  - This is an issue for manufacturers requiring a FIPS 140-2 certificate for their products
    - FIPS 140-2 expects the SP 800-57 guidelines to be followed for key management
- Re-use of SAKs after a membership change is risky
  - E.g., Stations A & B combine their KCs to create the same key before Station C joins and again after Station C leaves.

General MKA Design Goals

- There should be one IEEE 802.1af protocol, not separate pair-wise and group protocols.
  - Therefore a group key establishment protocol is needed
  - But it should be efficient for 2 stations and 2-*n* stations, where *n* is O(20)
- Must provide *n*-way anti-replay protection, and allow any station to know whether or not another station is alive
- Must be able to install a SAK ASAP and report to upper layers that the link is up
- Must use conventional cryptography and key management
  - But as few cryptographic algorithms as possible so as to simplify the overall solution
- Ensure that there is always at least one valid SAK for the group.

Group Key Establishment models

- Key Contribution (e.g., KSP)
  - All stations provide keying material which is combined into a single SAK
  - This method is typically used for small groups, where the key must change when the group membership changes
    - This is to "key out" a group member when they stop contributing to the group key.
    - However, this is not possible with the IEEE 802.1af architecture, so use of a key contribution method is not compelling.
- Key Server (e.g., LKS)
  - One station independently chooses a key and distributes it to the other group members
  - Allowing for multiple key servers avoids a single point of failure

LKS Key Server Strategy

- One station on the LAN acts as a key server for the group
  - Each station in the group is a candidate key server.
  - The stations on the LAN at any particular time determine which of themselves is the key server.
  - If the current key server becomes non-responsive, the remaining group members elect one of themselves to be a new key server
- The election is not a separate protocol, but performed asynchronously by all stations
  - Each station should have the same group state available to it, and thus can use the same election heuristics

LKS Message Flow

- A broadcast message is used to convey information between group members
  - –Role of the sender (key server or not)
  - –Current SAK (if the message is from the key server)
  - –Peer liveness state

  NOTE: Other information, such as the most recently used IEEE 802.1AE Packet Number, should also be conveyed in the message. (I.e., the KSP LLPN/OLPN)
- Each station periodically sends a broadcast message, which provides a liveness proof and ensures that all stations have the up-to-date group state
  - Additional messages are broadcast as necessary

Key Replacement Events

- Policy (I.e. key lifetime)
- New member joins
  - Necessary to avoid the GCM security condition
- Member request
  - GCM IV is about to wrap
  - Should request it when it recognizes a new KS (to avoid the GCM security condition when swapping key servers):

KS1 ---> KS2 --> KS1

LAN/MAN Partitions

- If a LAN/MAN is partitioned, the group of stations in each partition will converge on a sub-group key server for the duration of the partitioning event
- When the partitions are re-joined, the entire group will re-converge on a single key server.
- This event is handled in the basic state machine.

LKS Cryptography (1)

- One 128-bit key (the CAK) is available for use by any MKA protocol.
- To avoid using a single key for multiple purposes, multiple sub-keys are derived from the CAK
  - All keys are defined to be 128-bit AES keys, and must be the same size
  - The AES cipher is used in Electronic Code Book (ECB) mode defined in NIST SP 800-38A.
    - It is safe to use when only one block is encrypted. E.g., encrypting one 128-bit data block to create a 128-bit sub-key
  - A Key Encrypting Key (KEK) is derived to obscure the SAK when transmitted over the wire

    KEK = AES-ECB(CAK, 0x0)

  - An Integrity Checksum Value (ICV) key is derived to provide to verify the integrity of a message

    ICV_KEY = AES-ECB(CAK, 0x1)

LKS Cryptography (2)

- The KEK is used to protect the SAK on the wire using the AES Key Wrap procedure defined in RFC 3394

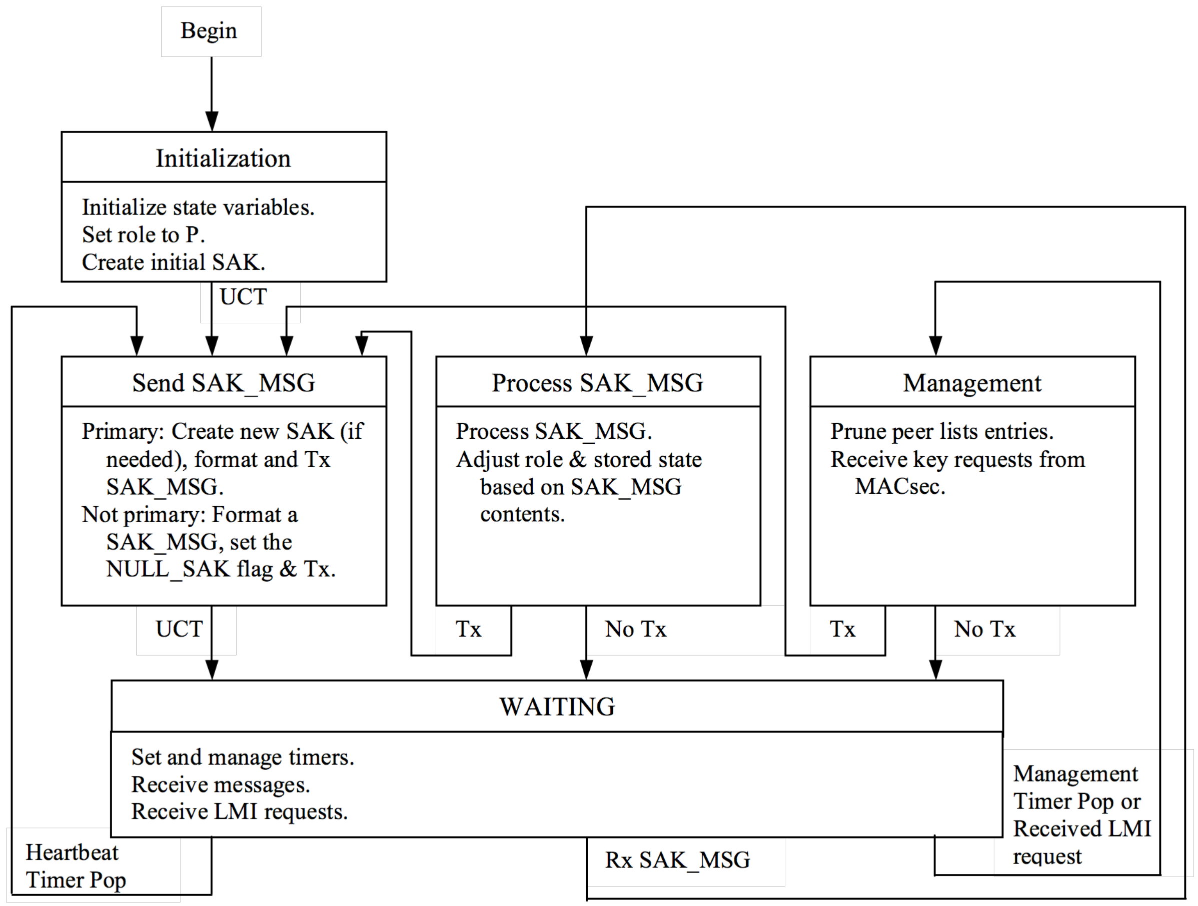
  ENCRYPTED_SAK = AES-KEYWRAP(KEK, SAK)

- The Integrity Checksum Value is computed using the AES cipher in Cipher-based MAC (CMAC) mode defined in NIST SP 800-38B:

  ICV = AES-CMAC(ICV_KEY,M,128)

- The SAK is generated using a strong Random Number Generator (RNG)
  - Note that a strong RNG is a necessary condition for FIPS 140-2, so this is not an onerous requirement.

LKS State Machine



Comparison of KSP/LKS

- LKS was designed to be similar to KSP in many ways
  – Liveness/Anti-replay method
  – Frame Format
  – Message Flow

KSP/LKS Anti-replay Method

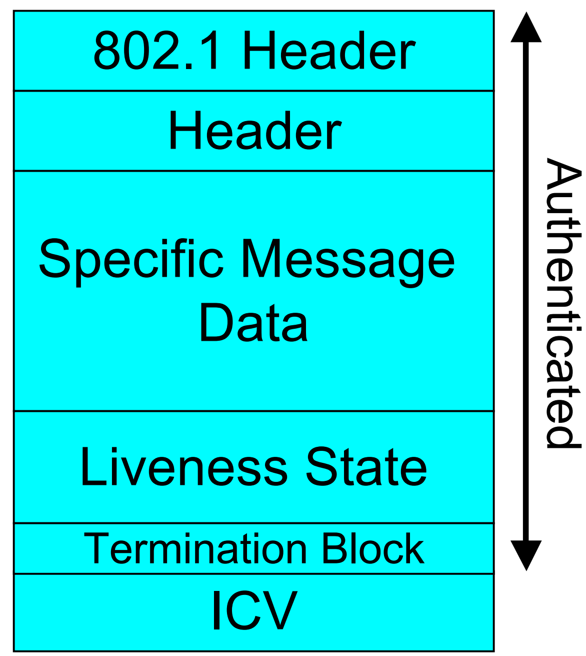
- Anti-replay
  - Each station maintains
    - A dynamic Member Identifier (MI)
    - A Message Number (MN) acting as a sequence number for messages including a specific MI
    - A list of peer MI/MN values
  - If a message is considered to not be replayed if
    - It has a known MI value, and
    - The MN value is larger than previously observed MN values.
  - When an MN approaches its largest value, the station chooses a new MI and resets the MN to 1.

KSP/LKS Liveness Method

- However, anti-replay is not sufficient
  - –The sender may choose a new MI value
  - –An attacker may delay messages
- Liveness builds upon the Anti-replay state
- Each station logically splits the peers into two lists
  - –A list of peer MI/MN values that are “live”
  - –A list of peer MI/MN values are are “potentially live”
- Each message includes the sending station’s “live” and “potentially live” peer lists
- A receiving station considers a peer to be “live” when it includes the receiving station’s recent MI/MN values in a peer list
  - – This proves that the station recently received a message from the station, and thus is “live”.

KSP/LKS Frame Format

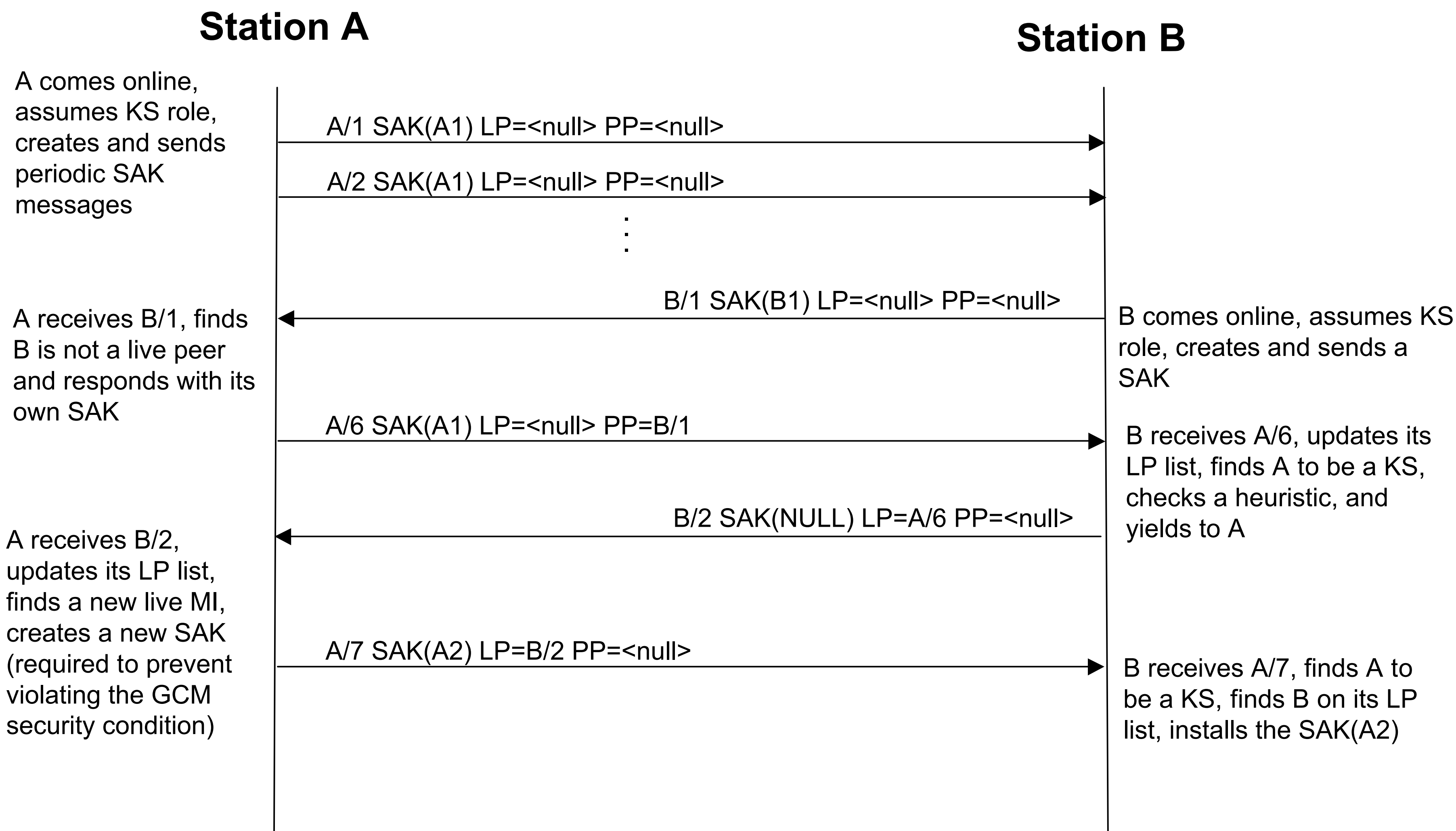
- LKS message fields
  - IEEE 802.1 Header (with a new Ethertype defining MKA).
  - LKS Header (station identity, etc.)
  - Specific Message Data (e.g., SAK)
  - Liveness State (Live Peer and Potential Peer lists)
  - ICV field
- Specific Messge Data differs
  - KSP includes a Key Contribution (KC)
  - LKS includes either
    - An encrypted SAK (KS)
    - Flags denoting that no SAK is present (non-KS)



Message Flow

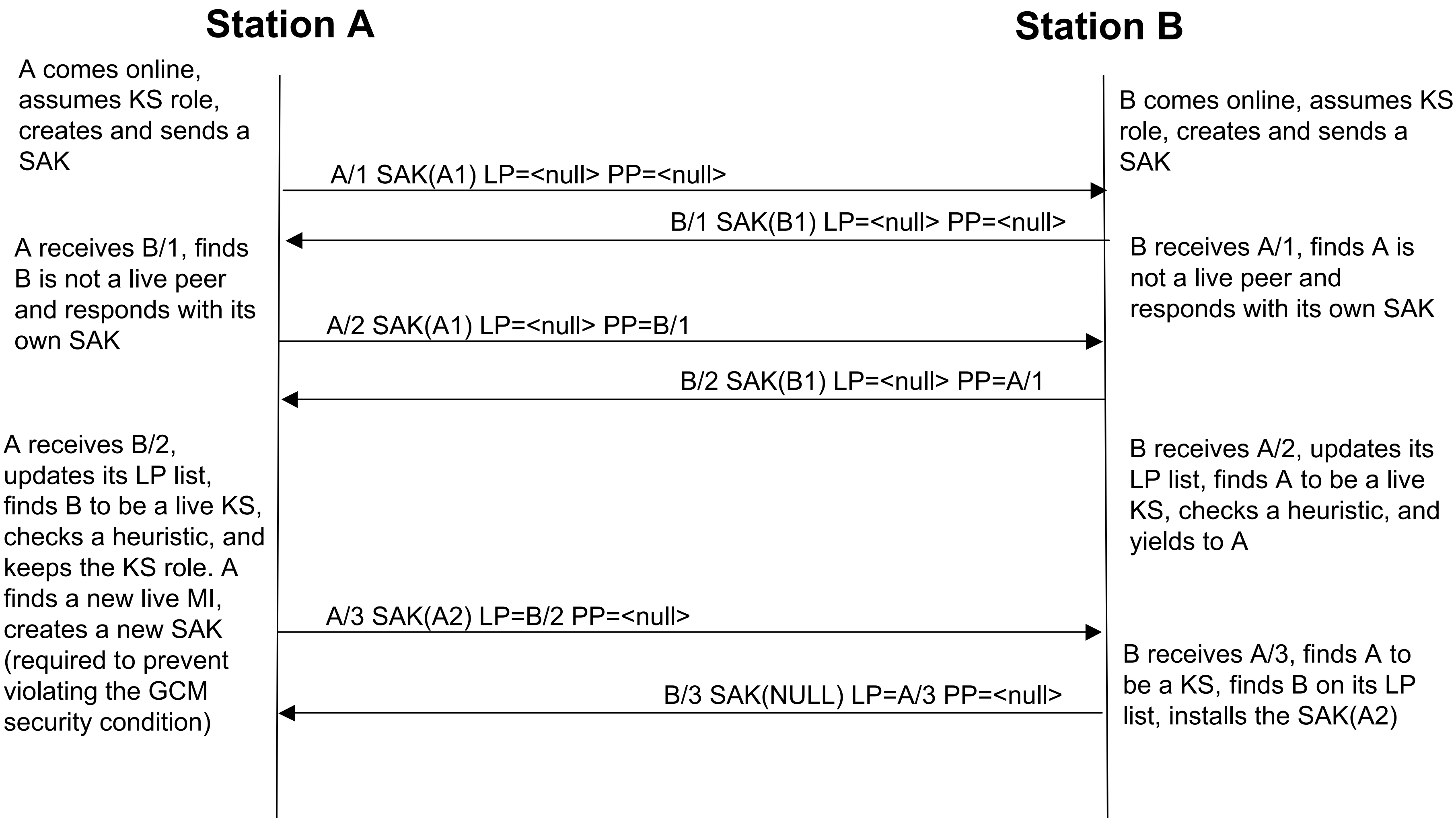
- The liveness requirement necessitates periodic broadcasts
  - This also helps quickly adjust to partition/join events
- When a new SAK is created it may be useful to send an immediate message rather than wait until the end of the period.
  - This requires no change to the state machine.

LKS Example 1:
Peers come online sequentially



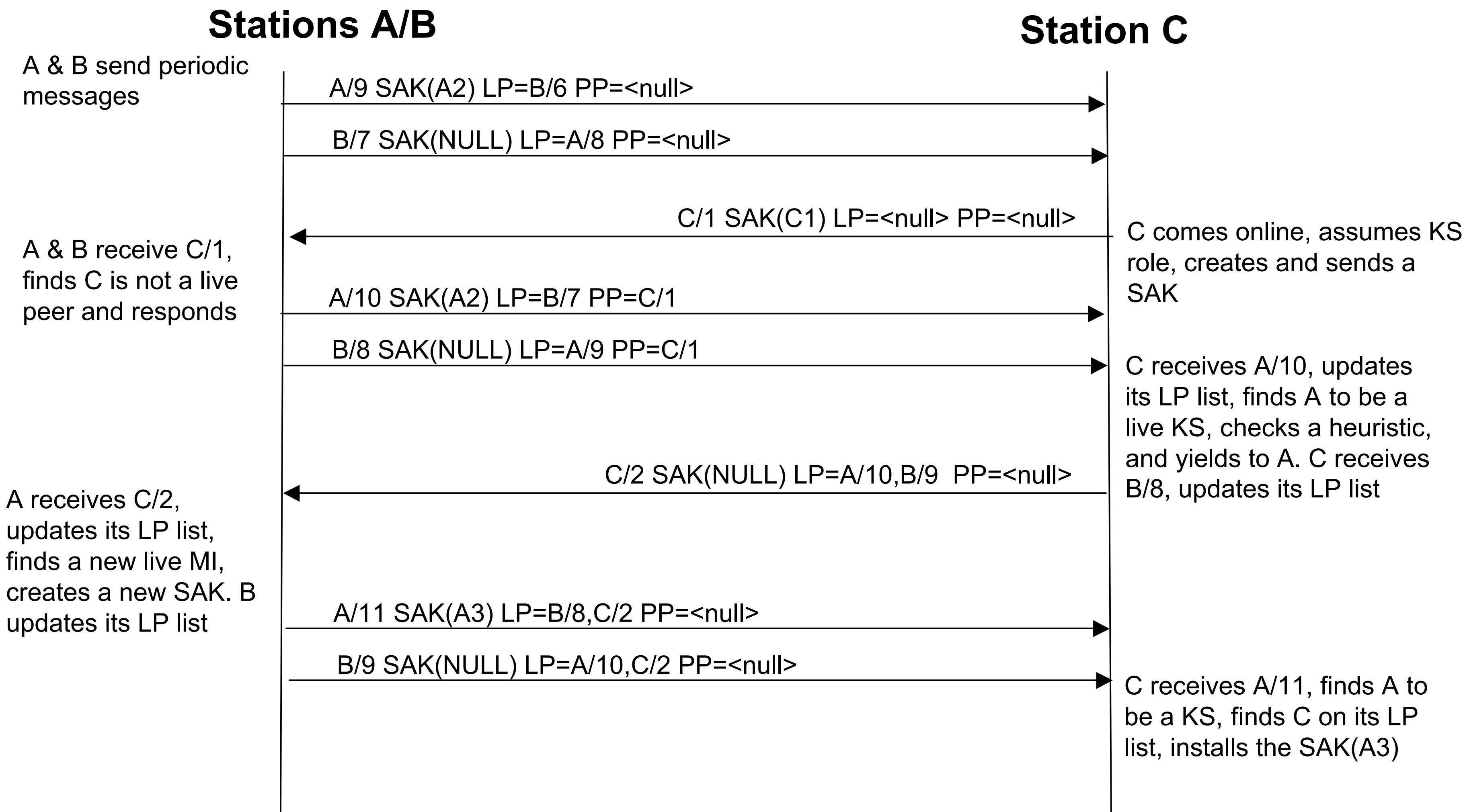
LKS Example 2:

Peers come online simultaneously



LKS Example 3:

Adding a third peer



Enabling per-user keys

- Key distribution allows an optional model where each device develops and distributes its own key
- Benefits:
  - Avoids issues surrounding the GCM security condition
  - Each sender controls the duration and usage of its own key independently.
- Issues:
  - Every station needs a key slot for every machine on the LAN, and must be prepared to switch to group mode if needed.