MKA optimization for group CAs

MACsecKey Agreement (MKA):

Existing procedures and possibilities for large groups and fast startup.[1]

Mick Seaman
mickseaman@gmail.com

1. See https://www.ieee802.org/1/files/public/docs2025/x-seaman-mka-optimizations-0325-v04.pdf for detail. This presentation follows that note almost exactly.

Agenda

1. Overview
2. Selective MKPDU validation
3. Repeated MKPDU transmission
4. SAK (data key) distribution
   - 4.1 Initial SAK distribution
   - 4.2 Follow up and rollover
   - 4.3 Requirements and rules
   - 4.4 Random Number Generation (RNG) requirement
   - 4.5 Installation and use (effort and timing)
5. Rapid Group CA formation
   - 5.1 Multiple initial participants
   - 5.2 Multiple participants join an existing CA
   - 5.3 Prior knowledge
   - 5.4 Pre-distribution
6. Continued SAK distribution

Selective MKPDU validation

MKPDUs ‘integrity protected’ not ‘confidentiality protected’

- Allows network administrator to monitor protocol without knowing secret keys (which can be held very closely)
- Receiver can prioritize (delay) or omit validation and use of non-time critical MKPDU information
- Safe under existing threat model (attacker can selectively prevent the delivery of any frame)

Caveats:

- Duplicate MI detection
- Maintaining liveness
- Confirming connectivity
- Can increase total number of MKPDUs transmitted
- Peer determination (can’t just listen to Key Server)

Repeated MKPDU transmission

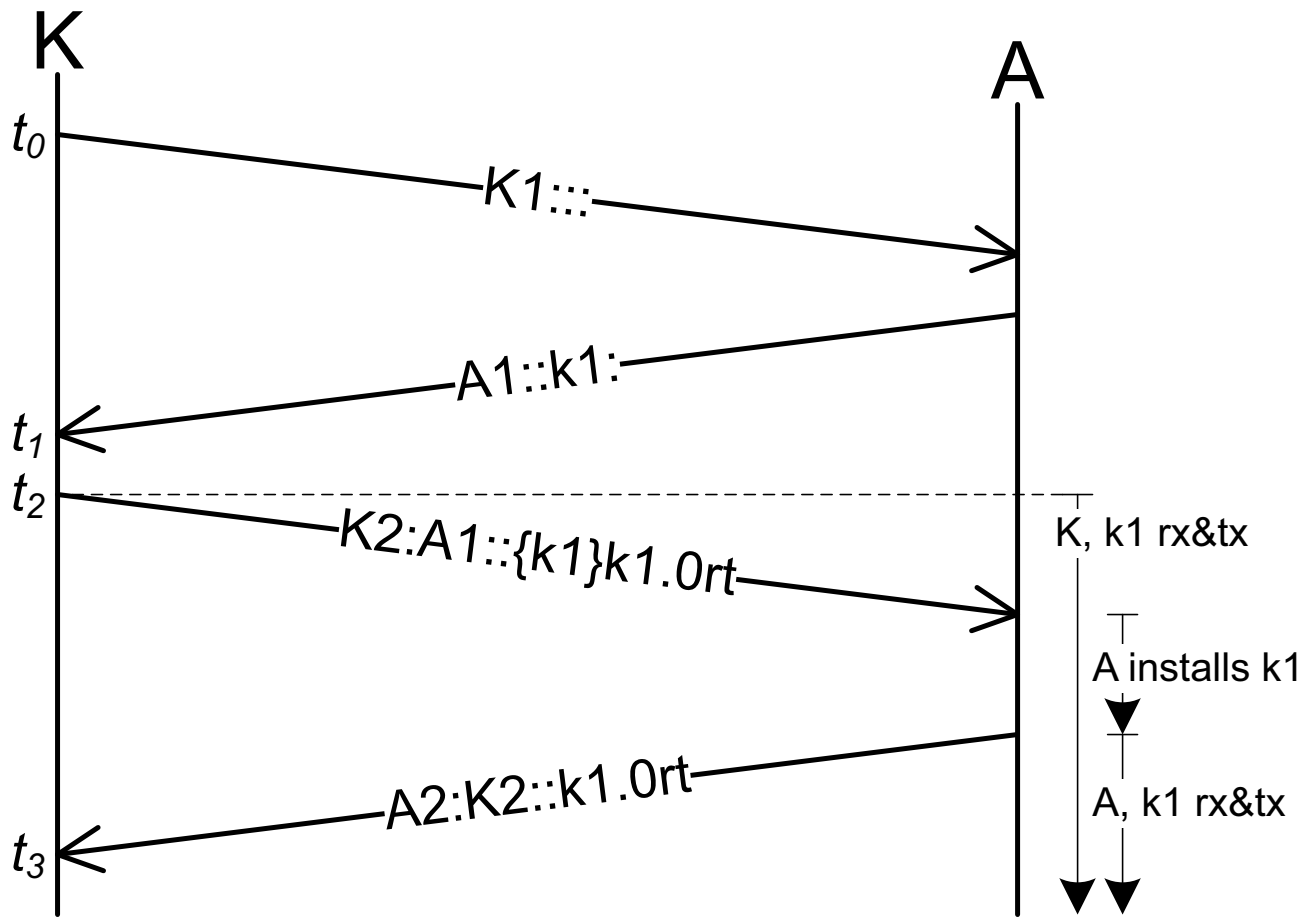
An MKPDU can be repeated ‘as is’

- Safe under existing threat model (attacker can copy any frame, and can transmit arbirary frames)
- Key Server can use to rapidly poll for new participants as they power up
- Rapid polling can facilitate addition to Key Server’s Potential Peer List, prove Key Server liveness to new participants, and thus rapid addition to Key Server’s Live Peer List

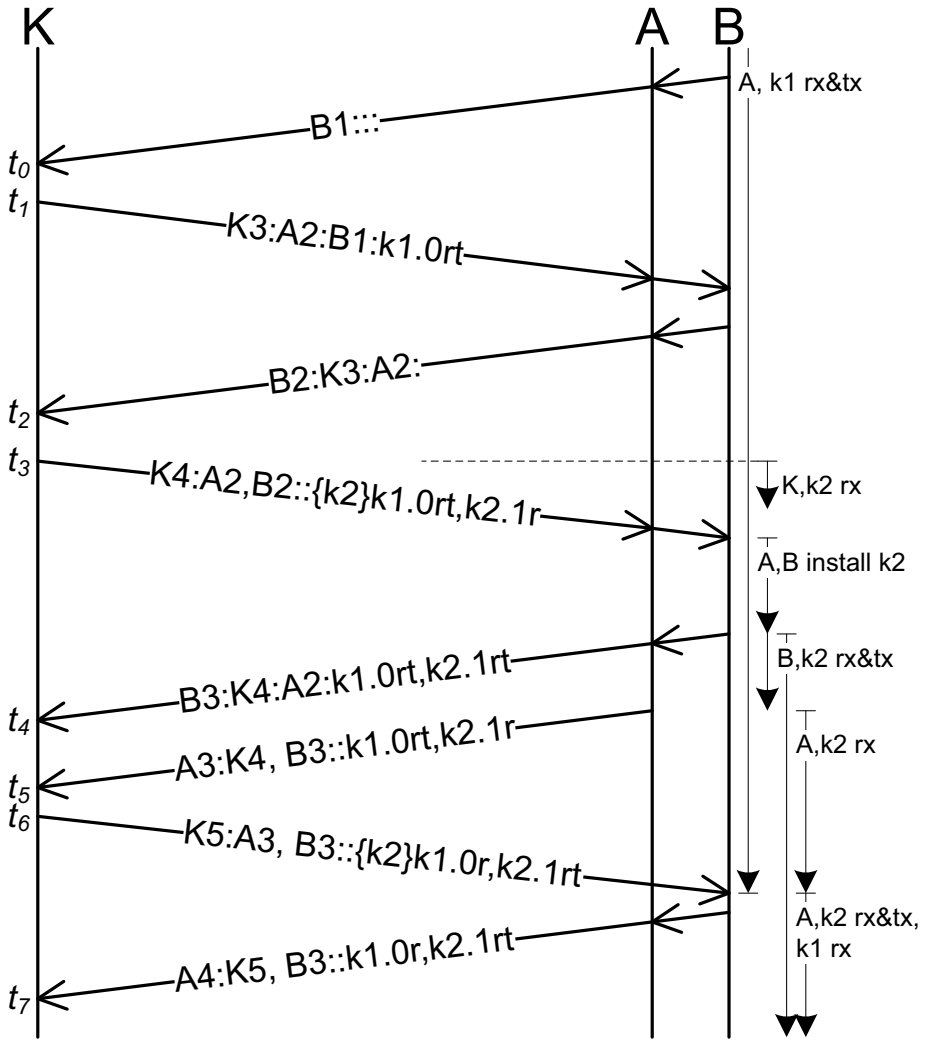
Caveats:

- Could overload participants that don’t selectively validate

Initial SAK distribution (example)



SAK distribution—follow up and rollover



SAK distribution—requirements, rules

Requirements:

- Ensure SAK.nonce not repeated for different data frames
- Ensure different participants do not share nonce
  - Key Server's Live Peer List in MKPDU with Distributed SAK determines who can use SAK
- Fresh SAK on reauthentication/fresh CAK for PFS
- Fresh SAK also guards against:
  - replay of old data to new participant
  - SAK extraction from compromised equipment removed from network

Current rules:

- Fresh SAK on change in Key Server's Live Peer List
- No distribution until Key Server's Potential Peer List empty or 6 seconds since last SAK distributed
  - when participants become available at roughly the same time, a little patience will result in initial SAK distributed to all

RNG requirement[1]

Each MKA participant uniquely identified by randomly chosen 96-bit Member Identifier (MI)

- No reliance on MAC Address uniqueness, guarding against:
  - Sloppy vendor allocation
  - Possible random allocation, with duplication risk
- Fresh MI required on participant restart, having forgotten possible prior SAK use

Implementors need to be aware of strong RNG requirement before MKA operation

- Key Server attempts to mitigate requirement with other restart detection heuristics susceptible to attack

1. RNG—a quality Random Number Generator.

SAK installation and use

1. Unwrapping AES Key Wrapped SAK receive in MKPDU
2. Calculation of any tables/intermediate results for Cipher Suite processing, install in h/w if applicable
3. Creation of SCI and AN indexed tables for reception, including per SC counts and per AN acceptable PN record
4. Calculation of any Cipher Suite-dependent values per received SA and per transmit AN
5. Enabling reception for SAK and SAs
6. Enabling transmission

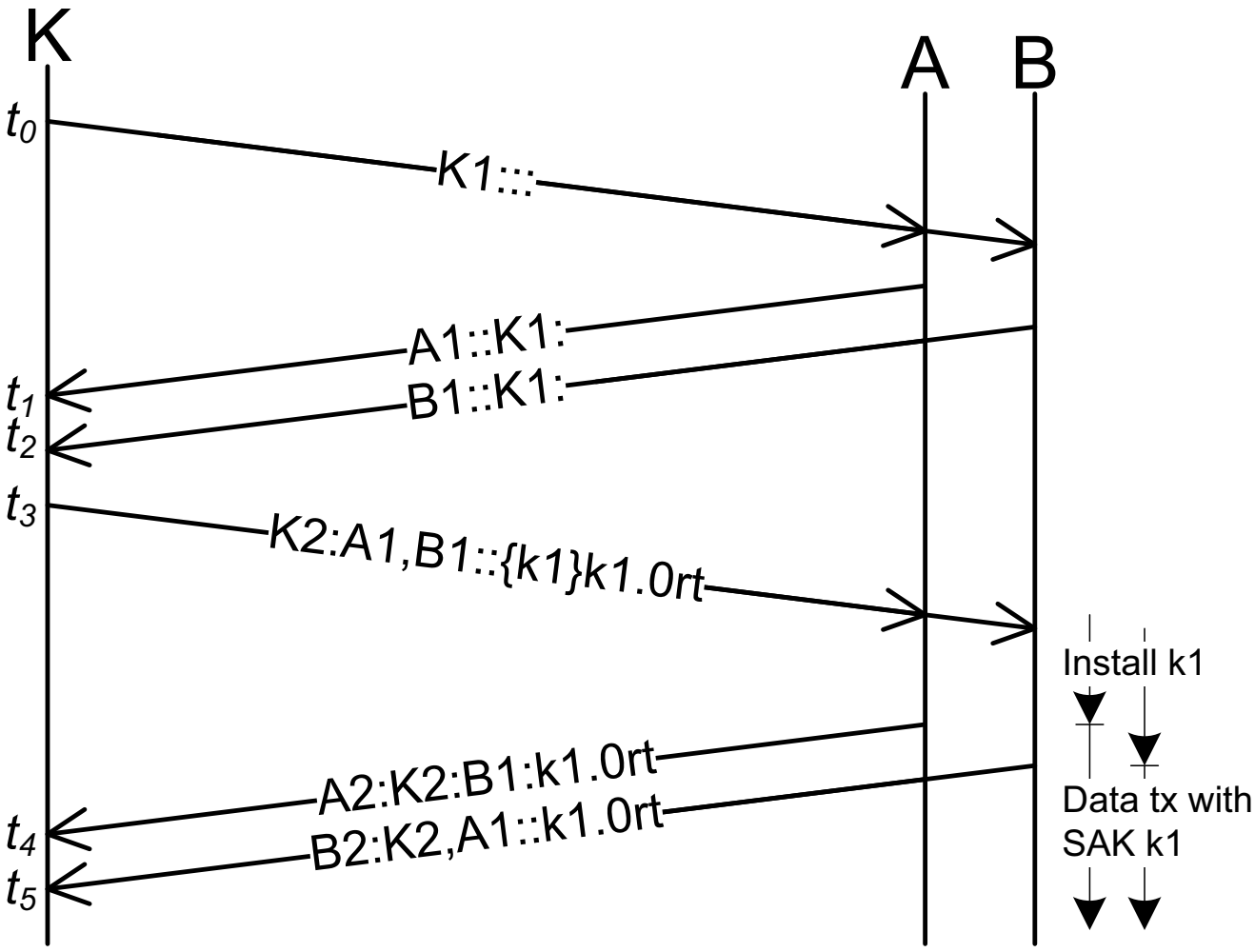
1 & 2 may be the most (implementation dependent) effort

1–4 can be completed, risking wasted effort, unconditionally

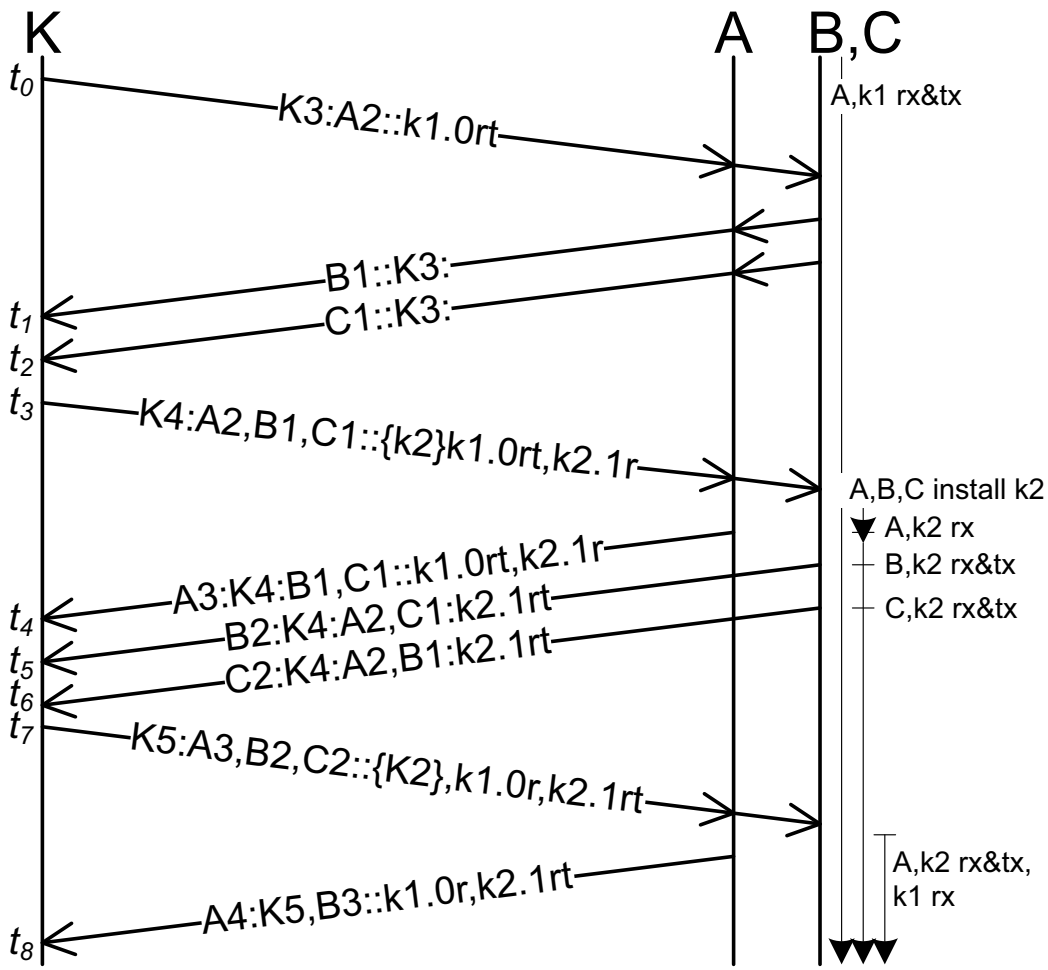
5 only if Key Server known live, risks replay otherwise

6 only on Key Server prompt, selecting CA participants

Multiple initial participants form a CA



Multiple participants join an existing CA



Prior knowledge

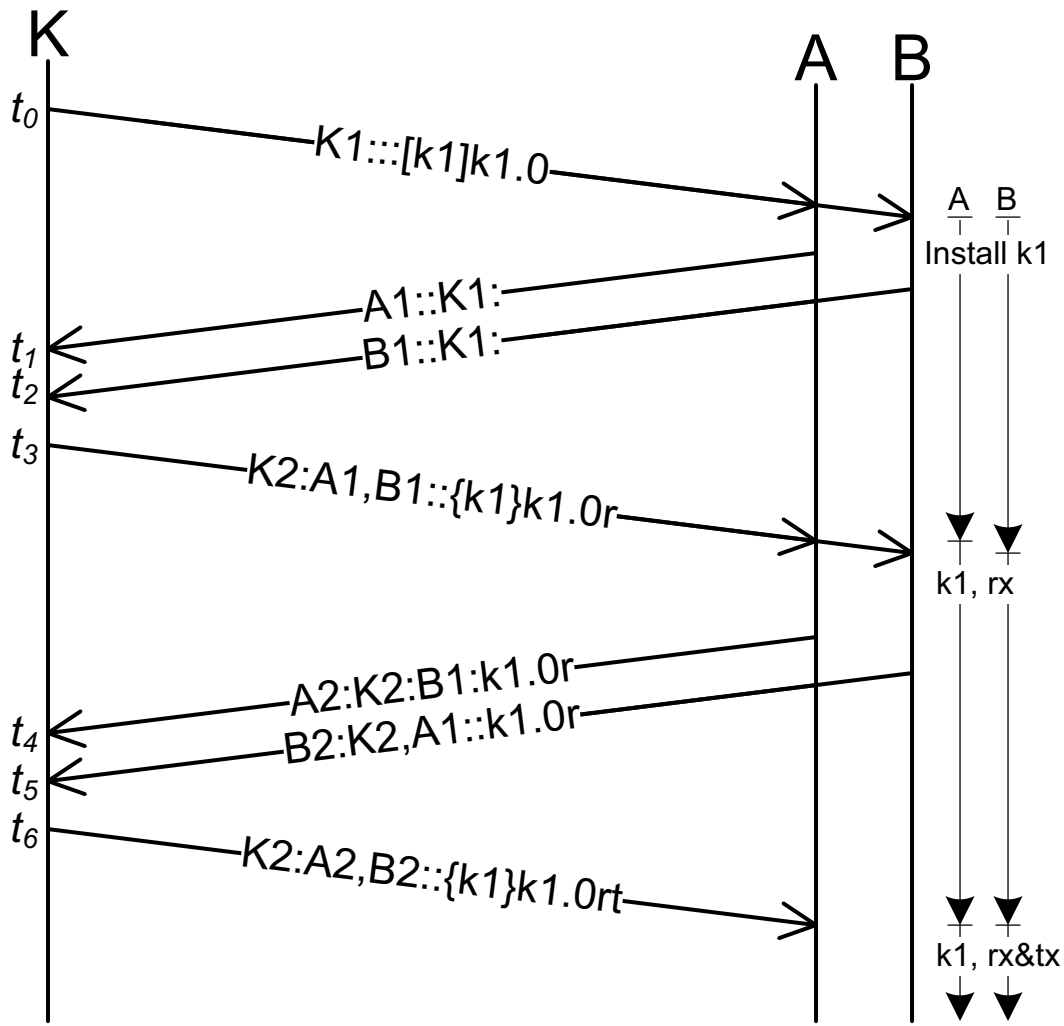
Rapid Group CA formation helped by Key Server knowing:

- Expected time for last participant to become available; or
- Expected number of participants for viable system operation; or
- Identity (MAC Address) of each essential participant.

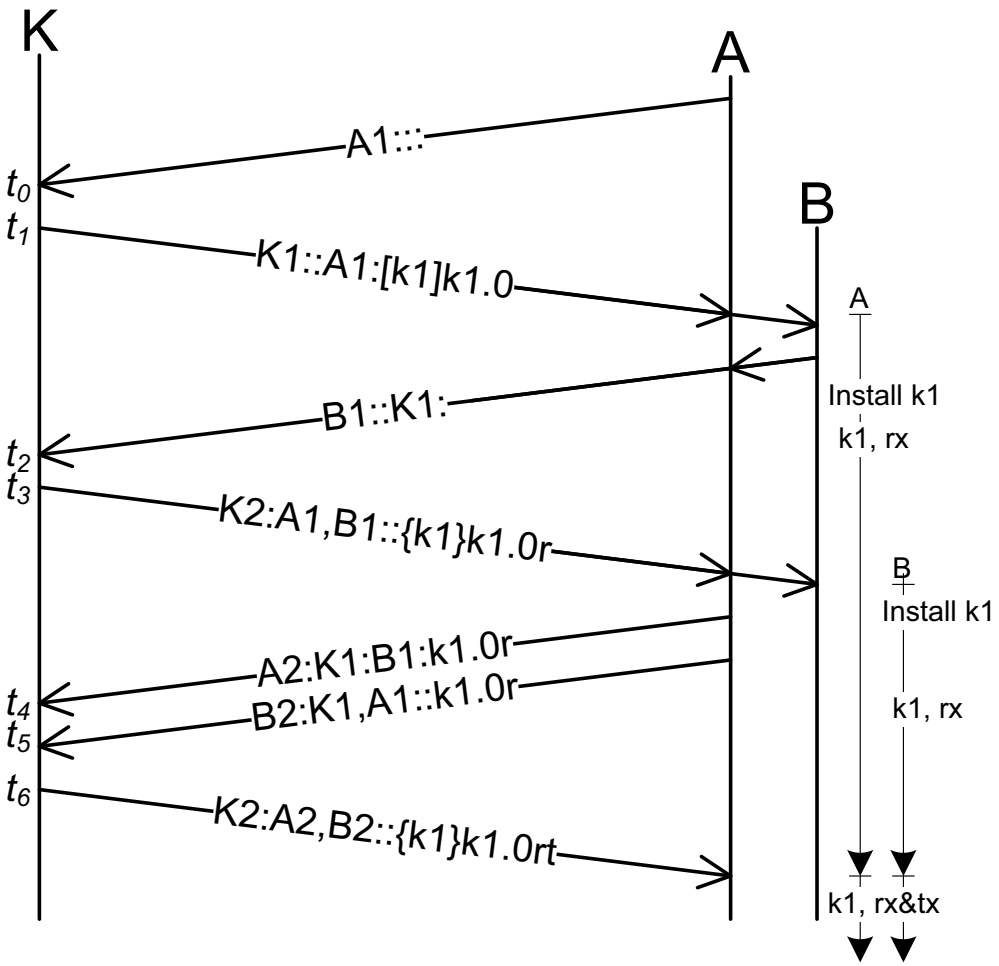
Rule for fresh SAK distribution (not until Potential Peer List empty) helps delay until all participants available.

- Needs cut off time if partcipant restart anticipated.

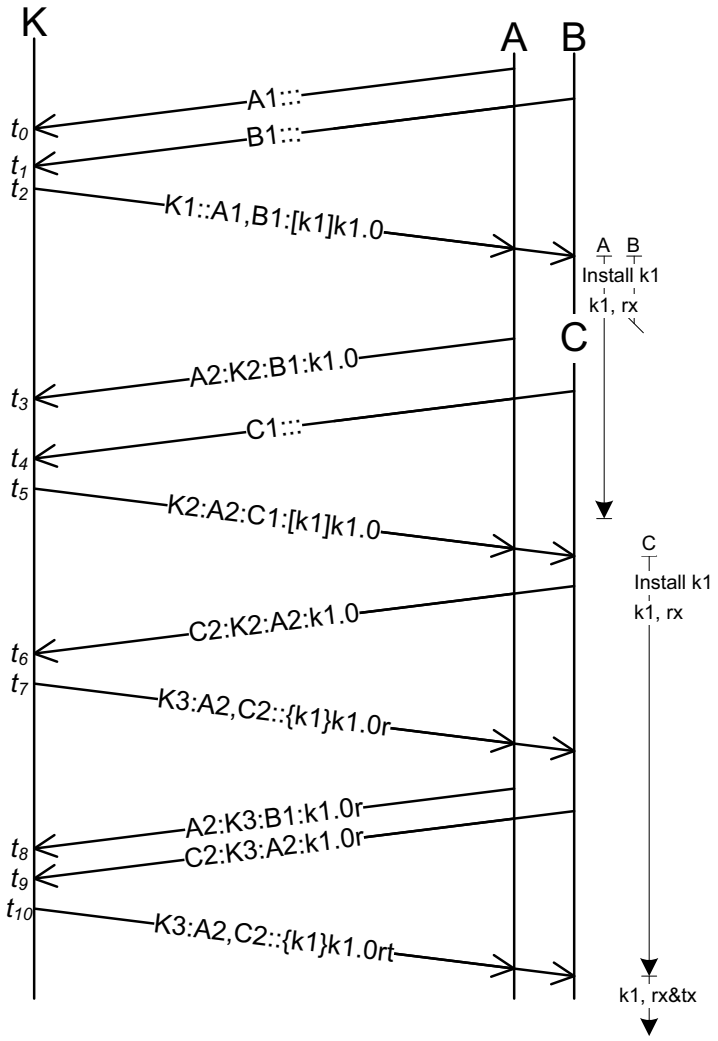
Pre-distribution



Pre-distribution (2)



Participant restart



Continued SAK distribution

6. Continued SAK Distribution

Cipher Suite specific distribution of a previously distributed SAK to a change set of participants, made possible by SAK.nonce uniqueness.

- 6.1 Lossless considerations
- 6.2 New vs restarting participants (PN Cipher Suites)
- 6.3 Nonce extensions
- 6.4 XPN rollover
- 6.5 PN nonce extension with KN rollover
- 6.6 PN nonce extension without KN rollover
- 6.7 SSCI nonce addition

Lossless continued SAK use

Key rollover, from one SAK to its successor, is lossless.

Key Server transmits with new SAK when all others can receive, others when Key Server transmits.

Same rollover, with Key Number increment, can be used when transitioning from SAK.nonce to SAK.new-nonce

- Less effort w/o rollover, but unclear when communication posssible between existing and new participants, and between new particpants
- w/o rollover, new participants have increased exposure to data replay

Background: When a station's interface is 'OperUp', the station should be able to communicate with its peers. Historically not always true—'interface up' can reflect MAC electronics, not intermediate software. When security is added, protocol exchanges are needed before the interface is truly up. Upper layer configuration protocols may not expect differences between unsecured and secured links, but stations can't exchange frames to 'OperUp' with the precision provided by media access control methods. To avoid initial data frame loss and timeout for protected configuration protocols, a point-to-point interface should be able to receive frames if it can transmit (so the second peer to come up can successfully start higher layer protocols without initial timeouts). A multipoint-to-multipoint interface should be able to receive from any other station attached to the LAN if it can transmit.

New vs restarting participants

Existing non-XPN Cipher Suites nonce includes the SCI:

- Addition of new participant, with different SCI from any previously used with SAK, does not risk nonce reuse
  - Could rollover Key Number, with notional reinstallation at existing participants to cover connectivity gain signalling
  - With or without key rollover, new participants more exposed to data replay
- Restarting participant forces new SAK

Nonce extension possibilities

XPN Cipher Suite KN rollover (6.4)

- KN is already part of Salt, participants can change and repeat SSCIs when KN changes, minimal standard change

PN nonce extension with KN rollover (6.5)

- Encode KN (or part) in largely unused existing SCI Port ID in SecTAG avoiding MACsec protection/validation change
- Formally define as new Cipher Suite, so capability known, and use explicitly selected

PN nonce extension w/o KN rollover (6.6)

- Key Server allocates participant specific Port ID additions (?)

XPN SSCI nonce addition w/o KN rollover (6.7)

- Allocate SSCIs in order of appearance in Key Server's Live Peer List, reserve MI value for null entries (past or restarted participants)

XPN rollover

PN nonce extension with KN rollover

PN nonce extension without KN rollover

SSCI nonce addition