MKA Suspension

Brian Weis

The project scope of IEEE P802.1Xbx includes provision for “the ability to maintain secure communication while the operation of MACsec Key Agreement (MKA) is suspended”. It is common for Ethernet switches to support an In-Service Software Upgrade (ISSU) process, at which time the data services of the bridge are maintained while its software is reloaded. This paper proposes changes to MKA to support suspension during an ISSU event, but also maintaining the architectural integrity of MKA.

1 Introduction

MACsec Key Agreement (MKA) was first specified in Std. IEEE 802.1X-2010. MKA allows network equipment to securely exchange MACsec policy and keying material for the use of IEEE 802.1AE-2006 (MACsec). Bridges and end hosts (“stations”) authorized to communicate within a particular MACsec context share a Connectivity Association (CA), and use a shared long-term Connectivity Association Key (CAK) to protect MKA messages and distribute a MACsec key (SAK).

A basic protection of MKA is the identification of peers within the CA, and is proved by the use of the correct CAK in the construction of MKA messages. Another basic protection is the identification of “live” peers, which have also proven that they also recently processed an MKA message from the receiver station. A device accomplishes this by storing the Member Identifier (MI) and Message Number (MN) value pair taken from each MKA message that the station has recently seen, and then transmitting those pairs in subsequent MKA messages. When a receiving station sees its recently sent MI/MN pair in an MKA message from a peer, then it considers that peer “live”. It is important to note in this discussion that both the MI and the MN are dynamic values: the MI is a dynamic identity used within the MKA session, and the MN is a counter that provides replay protection for MKA messages sent using that particular MI value. Any station can leave and rejoin the MKA session simply by choosing a new MI and resetting the MN to its lowest value.

Once a peer has been determined to be live, a key server election is performed. The elected key server then distributes a SAK to all peers. When all devices report installation of the key in the Rx direction, the key server installs the new SAK in the Tx direction and begins using it. This is the signal to other stations in the CA to also install the SAK in the Tx direction.

Once a SAK has been installed, the MKA process on each station continues to broadcast MKPDU frames, typically emitted every two seconds. This has several purposes, but foremost among them is the determination as which peers are live (as previously discussed). The liveness property is particularly valuable in use cases with multiple peers (e.g., access port and provider networks), as the “dead” peer may be the key server. Should the key server die, the remaining devices perform the election process described above, and the new key server becomes responsible for distributing SAKs.

Each SAK has a lifetime, based on the number of MACsec-protected frames that the station has transmitted using the SAK. When any station determines that it needs a new SAK, it advertises this fact in an MKPDU. This causes the key server to immediately distribute a new SAK.

2 Problem Overview

Many Ethernet switches, particularly large capacity ones, implement data services entirely in blades or hardware logic independent from the supervisor software that manages state for the

data services. This supervisor software is responsible for providing the data services layer with updated state when needed. For the purposes of this paper, the data services state is composed of MACsec SAKs and associated policy. In a steady state condition the data services do not require intervention of supervisor software and can operate autonomously.

This possibility of autonomous operation of data services allows these devices to maintain data services even in the unavailability of the supervisor software. We call this situation a planned *suspension event*. One example of a suspension event is an In-Service Software Upgrade (ISSU), during which the data services run in a steady state while the supervisor software is reloaded. At the end of an ISSU event, the software may query the data plane for its current state, but in any case continues its mission of providing the data plane with the state it needs. Other suspension events may also take place, for example migration of control from a primary software supervisor to a backup software supervisor entity. Another happens when the data services blade itself performs ISSU while the software supervisor blade supporting MKA continues to operate but cannot transmit messages through ports until the data services blade resumes operation. For the rest of this paper these events will be simply called suspension events.

A planned suspension strategy (e.g., ISSU) is only effective as long as the data services state does not become stale before the supervisor software resumes. However there is currently no guarantee that MACsec state will remain stable during a suspension event. This is due to several factors of MKA design:

- Liveness requirements, where a station not responding within 8 seconds (defined in Std. IEEE 802.1X-2010 Clause 9.1) is assumed to be unreachable for both MACsec and MKA services.
- At any time a station may require a new SAK because its Packet Number (PN) counter has reached or exceeded the PendingPNExhaustion value and is at risk of being fully exhausted. This paper identifies this is a *PendingPNExhaustion event*. In order to properly re-key, all stations sharing the old SAK must be available to participate in the MKA exchange in order to receive the new key. Any station in the midst of a suspension event will not receive the new key.
- Use of the default Cipher Suite (and the cipher suite defined in Std. IEEE 802.1AEbn) with high-speed links (e.g., 100Gbps, 400Gbps) can reduce the lifetime of an SA to an order of 1 to 2 minutes. This may not be enough time for some suspension events to complete.[1]

The rest of this paper proposes a modest set of enhancements to MKA to allow stations to suspend communications within a CA to accommodate a period of time when they cannot participate within MKA. For a summary of the proposed enhancements, see Section 7.

3 Principles of Operation

To accommodate suspension of a single station within a CA, the following requirements should be met:

R1. A station is responsible for declaring its intent to suspend at least one MKA Life Time period prior to beginning its suspension.

R2. A station should inform other members of the CA an estimate of its suspension period, after which the station may be taken out of their live lists if it does not resume transmitting using the same identity. The use of a suspension period bounds the duration of the suspension and allows normal MKA operations to resume at a predictable time.

[1] The use of Extended Packet Number (XPN) cipher suites described in IEEE P802.1AEbw mitigates the short SAK lifetime problem. However, there is no plan to deprecate cipher suites not using the XPN method so short SA lifetimes remains a possibility.

[2] MKA Life Time period is used for the purposes of discussion and requires further. Using the existing MKA Life Time is convenient, however.

R3. When a station declares its intent to suspend, the Key Server may choose to distribute a new SAK before the beginning of the suspension period. Distribution of a new SAK is desirable as it reduces the likelihood of a new SAK being required while the member is suspended.

R4. Other stations (including the Key Server) are expected to honor the suspension, but there is no guarantee that either the SAK or membership of the CA will remain unchanged during its suspension. In other words, while the extensions outlined in this paper aim to increase the reliability of MACsec/MKA operation during a period of suspension, the suspending station cannot unilaterally enforce its policy on the CA.

R5. If a new station joins the CA during a suspension period, a policy should be mandated; either for the Key Server distribute a new SAK (effectively removing the suspended station(s) from being a live participant within the CA, or to delay distributing a new SAK until the suspended member proves liveness again (effectively postponing entry of the new station).

R6. A new SAK should always be distributed when a station declares a PendingPNExhaustion event, if the declaration happens when the Key Server is suspended.

4 Declaring MKA Suspension

A station preparing for an ISSU event declares suspension by including a new type of parameter set (See Figure 2) in an MKA PDU. The parameter set defines the *Suspension Period* for which it expects to be suspended. It does so at MKA Life Time prior to beginning the suspension[2], which comprises a *Suspension Pending Period*. The purpose for the Suspension Pending period is to allow the Key Server to deliver a new SAK before the member begins suspension.

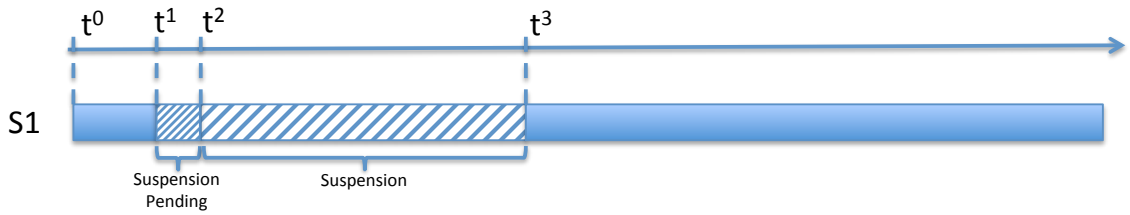

Figure 1. Suspension Period Definition

Figure 1 shows a single suspension event for station S1. S1 participates normally in the CA until its supervisor software requires a reload. S1 declares the event at time t^1, and actually suspends at t^2. S1 returns at t^3 after which it again participates normally within the CA.

4.1 MKA parameter encoding

In order for a station to declare a suspension it needs to include a new MKA parameter set in its MKAPDU messages prior to the suspension. An example of such a parameter set is shown in Figure 2. For the purpose of exposition this paper refers to is as a “Suspension Declaration parameter set”.

[2] MKA Life Time period is used for the purposes of discussion and requires further. Using the existing MKA Life Time is convenient, however.

Bit: 8	7	6	5	4	3	2	1	Octet:
Parameter set type = <New value declaring suspension>								1
RKEY	X	X	X	X	X	X		2
X	X	X	X	Parameter set body length				3
Parameter set body length (cont)								4
Suspension Period (in seconds)								5

Figure 2. Suspension Declaration parameter set

The flags are defined as follows:

- RKEY: The suspending station requests a rekey before the suspension period begins. This is only a request, however. The Key Server may choose not to cause a rekey to happen, particularly if a rekey recently occurred. Conversely the Key Server may rekey irrespective of the flag setting.

4.2 Resumption Semantics

Resumption is defined as the presence and ability for supervisor software to emit MKAPDUs following a suspension. Because there are various types of suspension events (see Section 2) more or less MKA state[3] will be available to the station suffered an outage. Following are some possible scenarios.

4.2.1 Minimal Available State

At minimum, the station must have the previously used CAK available. This is most likely to happen when the software managing MKA reloads and has only the cached CAK (as specified in Std. IEEE 802.1X-2010 Section 12.6). It can therefore begin to participate in MKA again after first choosing a new MI and re-establishing liveness with the rest of the stations in the CA with this new identity. This is a normal MKA message flow when a new station joins the CA.

Notes:

1. If there were no other changes to group membership, then the resuming station will have retained the most recent MACsec state, and data services will have continued even though MKA was not available. So from the point of view of the resuming station, the suspension had the expected result. The identity change after resumption is inconsequential to the suspending station.
2. It may be that the MKA code cannot recover MACsec state from hardware. I.e., LMI interfaces may not be sufficient in order for MKA to re-create its state. In this case, MKA may prematurely overwrite these “orphaned” in-use SAs and SAKs when installing the next set of SAs (rather than moving the current SAs to old SAs, which is the usual semantics for avoiding the loss of data). This could result in a short outage on the resuming station.
3. Because the station does not use it’s original MI value, peers (including the Key Server) will not detect that it has returned before the end of the Suspension Period it originally declared, and indeed will not differentiate between a resuming station and a new station. This has an effect on Key Server policy of distributing a new SAK, and will be discussed in later sections of this paper.

[3] I.e., the PAE management information, see Std. IEEE 802.1X-2010 Figure 12-3.

4.2.2 Complete Available State

In some suspension scenarios the software executing MKA retains all state. This can happen if the software executing MKA did not suspend, but other portions of the system suspended and removed its ability to transmit MKAPDUs. Alternatively, the software may have cached its Participant state (as shown in Std. IEEE 802.1X-2010 Figure 12-3) or the suspension may be the result of control switching between a primary supervisor instantiation and a backup supervisor instantiation. In each case, the station can resume MKA using its prior identity and to act as if it had never left, except that the MN for each MI reported in its MKA message may not match the current definition of "acceptably recent MN". However once it again has the ability to transmit MKAPDUs, within MKA Life Time seconds[4] after resuming, the station should be able to prove liveness again.

5 Processing a Suspension Declaration

Stations within a CA (including the Key Server) will observe the Suspension Declaration parameter set in a suspending station's MKAPDU, which requires some modest changes to their liveness checks.

5.1 Determining liveness

Clause 9.4.3 of Std. IEEE 802.1X-2010 states "Peers are removed from each list when an interval of between MKA Life Time (see Table 9-3) and MKA Life Time plus MKA Hello Time has elapsed since the participant's recent MN (see above) was transmitted." For suspended peers, the semantic should effectively become the following (changes in italics): "Peers are removed from each list when an interval of between MKA Life Time (see Table 9-3) and MKA Life Time plus *the Suspension Period* has elapsed since the participant's recent MN (see above) was transmitted.

A station should mark each peer on the live list that has declared a suspension period and store enough state to judge when the suspension period has completed for that peer. The information stored is implementation specific based on its timer strategy, but it should be noted that it is possible to perform this processing without defining a new MKA timer value. This extension to MKA does indicate the definition of a new "MKA Suspension Period" variable Timeout value in Std. IEEE 802.1X-2010 Table 9-3.

5.2 Basic Resumption Operation

Stations receiving an MKA event with the Suspension Declaration parameter set take note of the Suspension Period in the parameter set and initialize their Suspension Period for the station to that value.

The Key Server may immediately distribute a new SAK upon receipt of a Suspension Declaration parameter set. It will only do so if its policy indicates it should, and/or if the Suspension Declaration parameter set included the RKEY flag and it chose to honor the request. This is a conventional rekey event needing no further discussion.

Because the suspending station may return with either minimal policy or complete policy, peers will see different conditions, described in the following sections.

5.2.1 Station Resumes during the MKA Suspension Period

If the station proves liveness with the same MI before the Suspension Period completes, then the station is assumed to have returned from suspension with all MACsec state intact, and the station is marked in the local live list as no longer being suspended. A Key Server observing a station proving liveness within its declared Suspension Period does not distribute a new SAK, because the live list did not change.

[4] Or as short as MKA Hello Time seconds if there are no packet drops.

5.2.2 Station Does not Resume

If a suspended device resumes using a new identity (as described in Section 4.2.1), it cannot be differentiated from a new station joining the group. According to existing semantics, this will require the immediate distribution of a new SAK, followed by the members of the CA transmitting using the new SAK. Rekeying a new member while an existing member is suspended could result in a data services outage for the suspended member. However, in a CA with stable membership it can be assumed that the new identity does belong to the resuming member and the immediate key change will do no harm. Therefore, in the basic resumption case we can see that existing Std. IEEE 802.1X-2010 rekey semantics remain fully applicable.

When the Suspension Period completes, the suspended station's MI is removed from the live list.

6 Scenarios

Because MACsec topologies and use cases are varied, a variety of scenarios stressing the above suspension semantics are possible. This section describes some of those scenarios, beginning with the trivial case (see Figure 3). However, because each station in a CA acts independently, it is possible for multiple stations to suspend in a non-coordinated fashion. Figure 4 shows a set of stations participating in a CA over time, and will be used to illustrate several overlapping suspension scenarios.

6.1 Trivial Resumption Scenario

A trivial resumption scenario occurs when CA membership remains stable before and after the occurrence of a suspending station (see Figure 3). The trivial case is worth considering, as suspension is most likely to occur in infrastructure LAN use cases (e.g., Std. IEEE 802.1X-2010 Clauses 7.4 and 7.7). Infrastructure use cases typically comprise a stable CA membership, and also include only the infrastructure devices most likely to suspend.

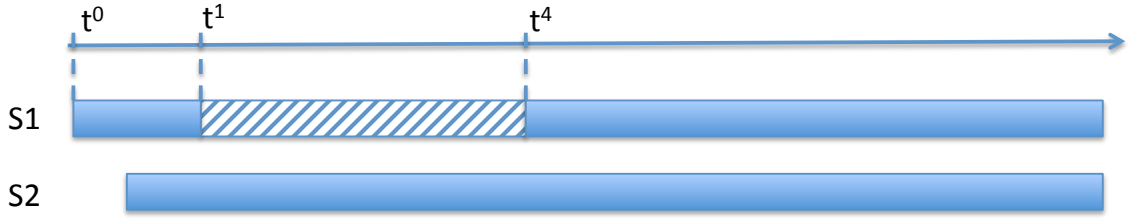

Figure 3. Trivial Suspension Scenario

The basis resumption operation (Section 5.2) will occur in the trivial resumption case.

6.2 Overlapping Suspensions: No change in CA membership

In this scenario, S1, S2, and S3 are live. Station S3 acts as Key Server, and S1 and S2 suspend in an overlapping manner as shown in Figure 4. Assume that S1, S2 and S3 share a SAK distributed prior to t^1.

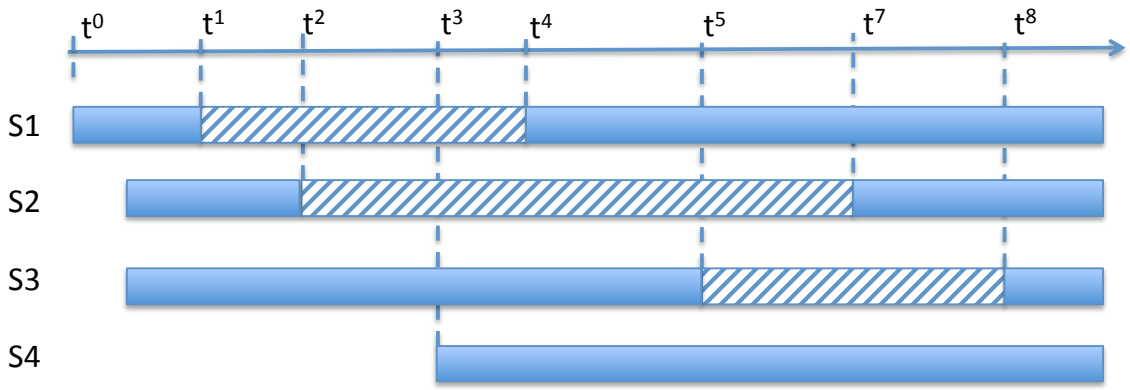

Figure 4. Stations with Overlapping Suspension Periods

Time t^4 is the critical juncture in this scenario, because S1 resumes when S2 remains suspended. If S1 retains its original MI value then it will resume as if it had not suspended (cases 1 and 2 in Table 1). There is no change in MKA state because the Key Server's live list did not change, and no new SAK is required. It can be observed that S1 will soon learn that S2 is no longer responding and take it off of its live list. Because the Key Server continues to declare S2 as live in its MKA messages, S1 will put S2 on its Potential Live list (representing peers that may become live peers in the future). But because the Key Server's live list does not change it will not distribute a new SAK. Whether or not S2 retains its MI does not matter, because all members are live again by t^7 and an ordinary rekey will happen as usual.

There are additional considerations when S1 takes a new MI value. Assuming Std. IEEE 802.1X-2010 rekey semantics (as described in the Trivial Resumption Scenario), the Key Server will rekey immediately when a new MI value is observed. Cases 3 and 4 of Table 1 shows the resulting outages suffered by S2, and surprisingly the outage for S2 is perpetuated in Case 3 until the next rekey event. This type of outage is not acceptable.

Case	S1 retains its MI	KS Rekeys at t^4	S2 retains its MI	KS Rekeys at t^7
1	Y	N	Y	N
2	Y	N	N	Y
3	N	Y: S2 misses new SAK, drops frames until t^7	Y	N: S2 cannot recover because no new SAK was delivered at t^7
4	N	Y: S2 misses new SAK, drops frames until t^7	N	Y

Table 1. Overlapping Suspension Periods (IStd. IEEE 802.1X-2010)

A simple and reliable solution is to define a new MKA rekey semantic, whereby the Key Server forbears from rekeying until all suspensions have completed. For the purposes of exposition, this paper refers to this as an *All Resumed* policy. The only exception to this rule is to always rekey when one station signals PendingPNExhaustion, because it vital to provide a replacement SAK when one station is nearing the end of its Packet Number (PN) namespace. As shown in Table 2 each of the above issues is mitigated for S2. Indeed, this semantic can be shown to satisfy multiple overlapping suspensions (e.g., S1, S2, and S3) as long as all members hold the same SAK before the first suspension period begins.

Case	S1 retains its MI	KS Rekeys at t^4	S2 retains its MI	KS Rekeys at t^7
1	Y	N	Y	Y
2	Y	N	N	Y
3	N	N	Y	Y
4	N	N	N	Y

Table 2. Overlapping Suspension Period: No SAKs until end.

This semantic would be sufficient for most infrastructure cases that would take advantage of a Suspension Period, because most likely the CA is composed of a pair of stable bridges (e.g., S1 and S2).

6.3 Overlapping Suspensions: Addition of stations during suspension

The previous section held the assumption that all stations held the same SAK at the beginning of the first suspension period. When this assumption is relaxed additional problems appear.

Consider the previous case of S1, S2, and S3 holding the same SAK distributed prior to t^1, followed by the addition of S4 during the suspension of S2. Using the All Resumed policy, S4 would not acquire a SAK and begin emitting MACsec protected data services until t^8 along with S1, S2, and S3. This could be an observable delay to users and administrators.[5]

It is tempting to revert to a policy of allowing a rekey at t^3 (i.e., when S4 joins the CA). However, recall that the appearance of a new station's MI value is indistinguishable to the key sever from S1 or S2 resuming with a new identity, and it has been previously shown that this is not an effective strategy for the suspended stations. There does not seem to be a clean method of adding new stations during a Suspension Period.

6.4 Suspended Key Server

At times, one of the suspended stations may be the Key Server for the CA. When the All Resumed policy is in place, this is inconsequential unless one of two events occurs:

1. The Key Server does not return from its suspension, or returns using a new MI value. Since the key server's MI will be removed from each station's live list, a new key server election will occur as usual and the new Key Server will immediately distribute a new SAK.
2. While the Key Server is suspended another station signals a PendingPNExhaustion event. This requires the immediate distribution of a new SAK, however it is unlikely that any member not in the Key Server role will be prepared to react to it. It would be possible to require all stations during a suspension event to note whether the Key Server is currently suspended when PendingPNExhaustion is declared and perform an election.

In either case, a suspended peer must not be chosen as the key server. Text in IEEE 802.1X-2010 Clause 9.5 that states "Each such participant selects the live participant advertising the highest priority as its Key Server whenever the Live Peers List changes, provided that highest priority participant has not selected another as its Key Server or is unwilling to act as the Key Server [as indicated by 9.5.1 b)]." This text should be amended with "… or is currently marked as suspended".

[5] To the extent that service packets such as Spanning Tree control messages are handled by the Controlled Port (CP) and thus cannot be transmitted until installation of a SAK, this port may not attract data packets. Of course, if the port is an infrastructure port located such that there are no alternate paths then the delay in user packets will be evident.

7 Conclusions

Certain events require a station participating in a CA to declare that while its data services will continue MACsec processing, MKA will be unavailable (suspended) for a period of time. This paper has shown that this can be accommodated with the following modest additions to MKA as defined in IEEE 802.1X-2010.

1. A new parameter set defining that the suspension will occur, and an expected timeframe for the suspension.
2. A modification to the algorithm determining liveness.
3. Addition of an *All Resumed* policy, which requires that (with the exception of an PendingPNExhaustion event) no new SAK will be generated until all stations on the live list have resumed.
4. Addition of a requirement specifying that when the Key Server is suspended, stations not in the Key Server role will perform an election if one station declares an PendingPNExhaustion event. The stations will not elect a suspended station, even if it has the highest priority.