**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | **Suggested Resolution Security Comments on 802.15.4i, Draft D05** | |
| Date | March 15, 2011 | |
| Source | René Struik  Struik Security Consultancy  723 Carlaw Avenue  Toronto ON  Canada M4K 3K8 | E-mail: rstruik.ext@gmail.com  Phone US: +1 (415) 690-7363  Phone Toronto: +1 (416) 466-8084  Cell: +1 (647) 867-5658 |
| Re: | IEEE 802.15.4 TGi/Draft D05 | |
| Abstract | This document provides suggested resolutions of security-related comments that were received with sponsor ballot on 802.15.4i/D05, as well as a number of suggested editorial changes to improve the security-relevant portions of 802.15.4i. | |
| Purpose | Assist with fixing/improving the security-relevant portion of the draft 802.15.4i standard. | |
| Notice | This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. | |
| Release | The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. | |

IMPORTANT NOTE:

The suggested resolutions in this submission are relative to IEEE 802.15.4i/D05. This document is an update of 15-11-188-01, which provided detailed security resolutions relative to 802.15.4i/D04, and simply transplants the suggested edits from D04 to D05 (to the extent that resolution of those comments inadvertently did not make it to 802.15.4i/D05).

* 15-11-188: suggested resolutions of security-related comments on 802.15.4i/D04;
* 15-11-189: actual textual changes suggested to 802.15.4i/D04;
* 15-11-190: suggested resolutions of security-related comments on 802.15.4i/D05;
* 15-11-191: actual textual changes suggested to 802.15.4i/D05.

**Comments on IEEE 802.15.4i, Draft D5**

COMMENTS BELOW - SUBMITTED WITH SPONSOR BALLOT RECIRC D05

1. [CID #19] (TR) Clause 7: Unfortunately, not all detailed changes that would have solved the security-related comments submitted during Sponsor Ballot on D04 that were deemed to have merit by the Sponsor Ballot Resolution Committee (witness 15-11-035-04) have made it to Draft D05 (as also seemed to have been acknowledged by the Technical Editor). This refers to Comments CID #71, CID #72, and CID #73 on Draft D04 in particular. For details of how (in my mind) the security clause should have looked like, please see 15-11-189-00 (which contains the end-result of changes to D04 text, including change markers) and the companion document 15-11-188-01 (which contains an in-depth rationale motivating changes to Clause 7 of D04 and a detailed breakdown of edits involved). The corresponding FrameMaker version of 189r0 is available from the commenter. **Suggested remedy:** Replace Clause 7 in its entirety by the text in 15-11-189, so as to fully address these comments. Note RS: documents 188r1 and 189r0 are relative to Draft D04; the translated version hereof, but now relative to Draft D05 can be found in documents 15-11-190 and 15-11-191, with FrameMaker version available as well, so as to minimize the workload for the Technical Editor.
2. [CID #20] (E) Clause 4.1, p. 7, l. 28-29: While I do appreciate that the unenforceable restriction on communication behaviour of RFDs (CID 67/D04) seems to have been removed in this draft, the resulting sentence flow misses some verbs. Replace “An RFD, on the other hand, is a device that not capable of serving either a PAN coordinator or a coordinator” by ““An RFD, on the other hand, is a device that is not capable of serving either as a PAN coordinator or as a coordinator”. **Suggested remedy:** Change accordingly.
3. [CID #21] (TR) Clause 6.4.2, p. 128, Table 52: According to this table, each device can only operate in at most one PAN (since there is not a set of macPANIDs; it is a singleton set or the empty set). This precludes deployment scenarios where a device may operate in two or more PANs at the same time (which could happen, e.g., if a device first is in a two-device PAN with a configuration device and then joins an “ordinary” operational network, with the configuration device once in a while still interacting with that device, e.g., for diagnostics or maintenance; another example would be a device transitioning from one PAN to the next without loosing “connectivity” [which Telco applications of 802.15.4 may wish]). The only solution right now is to replicate the entire MAC multiple times on a device and implement other dirty “MAC farm” tricks. The obvious way to facilitate the multiple PAN scenario would be to introduce a set of PANIds (and, for each PANId, potentially another short address, and coordinator), thereby facilitating truly distributed network management. **Suggested remedy:** Replace the PANId by a set of PANIds and corresponding other functionality, so as to facilitate network partitioning and transitionary effects. Obviously, this seems to be a big change (since also impacting third level filtering (Clause 5.1.6) and table organization, but conceptually this change is not major. This change may facilitate 802.15.4 conquering the world.
4. [CID #22] (TR) Clause 5.1.6.2, p. 41, l. 15-34: The current filtering procedure may accept frames originating from the receiving device itself (thus, providing looped behavior).This would allow an adversary to resend frames broadcasted by a device and have those accepted (via level 1-3 filtering) by that very device (hence, evidencing the looped behavior). Moreover, if this self-looped behavior is detected, one can implement all addressing related info, including that of the device itself, in one table with uniform processing (i.e., the macDeviceTable, cf. Table 58). **Suggested remedy:** With third level filtering, silently drop frames purportedly sent from the recipient device itself (this is a primitive level of “source address filtering”). Alternatively, this check can be incorporated in the incoming frame security procedure (Clause 7.2.3, prior to Step a), so that these frames are not silently dropped, but anomalies could be flagged to subsequent processes (by introducing a new status value indicating such looped behavior). In fact, with introduction of a new PIB parameter that indicates silently dropping this or flagging this condition, either solution can be facilitated. Note RS: I submitted this comment previously (CID #70/D04), which was rejected by the BRC, but now added an additional argument for this at the end of the comment.
5. [CID #23] (TR) Clause 5.1.6.2, p. 41, l. 24-27: the third level incoming filtering procedure includes a check as to matching of the destination address with the receiving device \*or\* a group to which it belongs. This suggest group addressing, which – to my knowledge – was never supported in 802.15.4-2006 (although some proposals have been made, cf., e.g., 05/180r0 and 02/474r2). This also causes problems in the security processing, e.g., with outgoing frame security processing (Clause 7.2.1), where the key to be used for securing the outgoing frame is looked up based on KeySource and KeyIndex, and where KeySource is a device (cf. Clause 7.4.3.1 and Clause 7.4.3.2). Thus, the concept of group addresses does break the current specification. **Suggested remedy:** Remove the group addressing implication in third level incoming filtering.
6. [CID #24] (TR) Clause 7.2.1, pp. 131-132: The current outgoing frame security procedure still does not check whether so-called “frame counter role-over” may have occurred. If frame counter roll-over indeed occurred, then reuse of the counter would completely break cryptographic security of crypto construct, due to nonce reuse (which is obviously not intended). **Suggested remedy:** implement this check via a corresponding Blacklisted element. For details, please see 11/189, Clause 7.2.1, Step l) and Clause 7.5.1 hereof (adding corresponding Security-Related MAC PIB attribute Blacklisted element in Table 59). Note RS: I did submit this as comment on D04 (CID 71/D04), which was AiP, but unfortunately not completely implemented with Draft D05; hence, the repeat. For full details, cf. document pair (11/188, 11/189) for details w.r.t. D04, and (11/190, 11/191) for those relative to D05.

Changes: (NOT UPDATED RELATIVE TO D05YET 0 WORK IN PROGRESS!)

* Clause 7.2.1: changed Step g), which now yields KeyDescriptor and allows indexing of Blacklisted element hereof later on.
* Clause 7.2.1: inserted Step g2) as Step h), which allows inspecting of Blacklisted element.
* Clause 7.2.1: inserted Step j2) as Step l), which sets the Blacklisted element if the frame counter has rolled over (this is not done for all other keys, since that was considered “nannying” by some TG4b people at the time, although it probably should, since each device has only one frame counter, irrespective of the number of keys stored). Note, however, that outgoing frame security with that frame counter will fail (Step f) from now on, unless the frame counter is reset by some TG4i-external process. Thus, all other keys could in theory be blacklisted via this external process then. Note RS: personally, I would rather include this safety precaution in the specification (TO BE DISCUSSED).
* Clause 7.2.2: The outgoing frame key retrieval procedure now yields a KeyDescriptor, rather than just a key, thus allowing subsequent inspection of, e.g., the Blacklisted element.
* Clause 7.2.2: This procedure now looks up the KeyDescriptor in two stages, via look-up of KeySourceDescriptor (a new procedure, inserted right after Clause 7.2.7 [DeviceDescriptor lookup procedure]), and by searching on the proper key index. This brings this more in line with the incoming frame security material retrieval procedure of Clause 7.2.4 and with the Key Identifier field.
* Clause 7.5, p. 154: introduced key source table.
* Clause 7.5.1, Table 58: added macKeySourceTable.
* Clause 7.5.1, right after Table 64: introduced Table 65 – elements of KeySourceDescriptor.
* Clause 7.5.8, p. 159: inserted Clause 7.5.8a, with description key source table.
* Clause 7.2.5: The KeyDescriptor lookup procedure now yields, indeed, the KeyDescriptor, if there.

1. [CID #25] (TR) Clause 7.2.4, p. 134: The current incoming frame security procedure still does not properly treat devices with so-called diplomatic immunity status (Exempt status). As an example, Step g) should happen prior to Step f), since if the frame was unsecured and the originating device has Exempt status, that frame may be accepted. With the current ordering, this frame may be rejected at Step f), since the recipient and originator may not yet share a key (which could happen if the device joins and is about to negotiate a key via some higher-layer protocol [this is the reason to have Exempt status devices in 802.15.4-2006]). In fact, there are more errors, since checking whether a device has exempt status involves obtaining the DeviceDescriptor, \*without\* at the same time obtaining the KeyDescriptor and KeyDeviceDescriptor (as currently all done at once in Step f). **Suggested remedy:** For details, please see 11/189, Clause 7.2.3 and impacted subroutines, as also fully explained in 11/188r1. Note RS: I did submit this as comment on D04 (CID 73/D04), which was AiP, but unfortunately not completely implemented with Draft D05; hence, the repeat. For full details, cf. document pair (11/188, 11/189) for details w.r.t. D04, and (11/190, 11/191) for those relative to D05.

Changes: (NOT UPDATED RELATIVE TO D05YET 0 WORK IN PROGRESS!)

* First some background: With 802.15.4-2006, an unsecured frame would be accepted if the security policy check would result in pass (security level allowed) or conditionally pass (security level normally not okay, but DeviceOverrideSecurityMinimum flag set). This is not granular enough, since it would allow any device to either by-pass normal security and not just the happy few that have explicitly been granted this diplomatic immunity status (what the Exempt flag in the DeviceDescriptor was supposed to achieve). As an example, if a device comes out of the box and tries and join the network without any keys, then this Exempt flag setting for association commands and data frames may be useful, so that higher layer protocols can set-up a key first (after which the diplomatic immunity for the joining device is revoked and it has to stick to the rules from now on). With TG4i/Draft D4, this is not the case any more (thanks to Step f). Nevertheless, one still has to dig up the Exempt flag of the DeviceDescriptor, without tieing this to keys, etc. (as the 802.15.4-2006 spec did), so as to check whether the specific originating device in question had Exempt status or not (this is what Step h does after the fixes below). If so, the unsecured frame would be accepted; if not, it would be rejected. The remainder of the incoming frame security processing (Step i) onwards) would then deal with the ordinary cases (no diplomatic immunity, real security applied, etc.).This requires untangling the procedure in 802.15.4-2006, 7.5.8.2.3, Step g, and getting a DeviceDescriptor independently of KeyDescriptor and KeyDeviceDescriptor. Thus, the comment and reference to seemingly massive changes (due to the untieing). In the end, this is not as bad as it seems, but may show as a big change bar nevertheless.
* Clause 7.2.3, p. 143, Step g, l. 25: the procedure in 7.2.5 is actually not the right procedure, since it has the wrong name and the output should be the DeviceDescriptor. Introduced the currently missing incoming frame device retrieval procedure as new procedure (cf. also 09/829r0, Clause 7.5.8.2.5).
* Clause 7.5, p. 156, Table 59: Removed the topmost two entries, since not required any more due to unentanglement of KeyDescriptor, DeviceDescriptor, and KeyDeviceDescriptor procedures. Similarly, one can remove Table 64 (roughly replaced by KeySourceDecriptor – cf. 09/829r0, Table 94).
* Clause 7.5, Table 61, p. 157: Removed the topmost two entries DeviceDescriptorHandle and UniqueDevice, since never invoked after edits. Inserted DeviceAddress instead (cf. 09/829r0, Table 91, p. 214).
* Clause 7.2.7, p. 146: This procedure should have as input the device lookup data and device lookup size, as determined in the incoming frame device retrieval procedure, and yield as output the DeviceDescriptor, if there. Made fixes as also indicated in 09/829r0, Clause 7.5.8.2.8.
* Clause 7.2.3, p. 146, Step k): Again, a mismatch here, since 7.2.4 is not the frame key retrieval procedure and output is KeyDescriptor and lots of other stuff. Corrected version would be the one of 09/829r0, Clause 7.5.8.2.4 (which was half-copied only, hence all the errors here). Note that the resulting Clause 7.2.2 and 7.2.4 are almost identical and use the same subroutines (thus, providing additional rationale for editing Clause 7.2.2).
* Clause 7.2.3, p. 146, Step l): Again, a mismatch here, since 7.2.7 is the not the KeyDeviceDescriptor lookup procedure and output is not a KeyDeviceDescriptor. Corrected version would be the one of 09/829r0, Clause 7.5.8.2.7. (which seemed to have been copied only half-way, hence all the errors here).
* Clause 7.2.6, p. 146: This procedure should be removed, since absorbed by the KeyDeviceDescriptor procedure (Clause 7.5.8.2.7 of 09/829r0) and the test on the Blacklisted element performed in Step l) of 7.2.3.
* Clause 7.2.3, p. 146: I moved the frame counter steps (Step i) and Step j)) to right after Step m) {the key usage policy check}. Motivation: with TG4e, one transforms the frame counter and requires the key (now in Step k), so to keep this all in a logical flow, it is better to change the order somewhat. This has no impact on behavior, but avoids cumbersome spaghetti code.

General note RS: please check reference to Annexes (right-concatenation, etc.), since now link is broken.

1. [CID #26] (E) Clause 7.2.7, p. 137, l. 31: The “Allowed security levels” parameter should be obtained from Table 62, which information is currently missing. **Suggested remedy:** change accordingly.
2. [CID #27] (TR) Clause 7.2.4, p. 134: The current (in Draft D05) security level checking procedure still has problems, despite (or perhaps due to) partial implementation of previous comment CID 72/D04). As an example, with Step h), if one ends up with a “conditionally passed status” and the originating device has diplomatic immunity (Exempt status), then the incoming frame should be accepted. However, it may very well be that the procedure that yields the DeviceDescriptor has failed, since entangled with obtaining keying-material related parameters (Step f). **Suggested remedy:** For details, please see 11/189, Clause 7.2.3 and impacted subroutines, as also fully explained in 11/188r1. Note RS: I did submit this as comment on D04 (CID 72/D04), which was AiP, but unfortunately not correctly implemented with Draft D05; hence, the repeat. For full details, cf. document pair (11/188, 11/189) for details w.r.t. D04, and (11/190, 11/191) for those relative to D05.

Changes: (NOT UPDATED RELATIVE TO D05YET 0 WORK IN PROGRESS!)

* Clause 7.5.1, Table 58: With description of *macSecurityLevelTable*, replace “minimum security level” by “set of security levels”.
* Clause 7.5.4, Table 62: Change this table to reflect a set of security levels, rather than a minimum. The corresponding SecurityModeDecriptors are inserted at the end of security-related tables (i.e., after Table 64 in draft). (With the current draft D4, the SecurityModeDescriptor is referred on in Clause 7.2.11, but missing.) Note RS: for now, I left the PIB entry *SecurityMinimum* in (even though this does not seem to make sense to me), since that was part of stated resolution of TG4i BRC conf call as of Fri February 11, 2011.
* Clause 7.5.4, p. 158: Replaced title “minimum security level table” by “security level table”, made corresponding change at beginning of Clause 7.5, and removed “minimum” from subsequent text.

1. [CID #28] (TR) Clause 5.1.6.4.3, p. 45, l. 20-28: The current text is somewhat ambiguous as to whether retransmissions involve not going through the outgoing frame security procedure again. Obviously, this is not intended, since that would result in sending a retransmission with updated frame counter and may result in receiving duplicates of the “same” frame. **Suggested remedy: C**larify this, since otherwise retried transmission that is simply the result of missing an ACK may result in accepting a frame twice at recipient's side (once with original counter value, another time with updated value). This is not intended behavior, but one can imagine some readers to construe this as such.
2. [CID #29] (TR) Clause 7.3.4.2, Table 53, pp. 149-150: The Auxiliary Security Header field is technically part of the MHR (cf., Clause 5.2.1, Fig. 30, p. 53) and it is obviously not the intention to copy that substring twice in the ‘a data’. Intention is as follows: Let STR be the right-concatenation of the MHR and MAC payload fields of the frame to be transmitted. The ‘m data’ now is the Payload field of the MAC payload in case the frame requires encryption and the empty string otherwise. The ‘a data’ is the unique string so that the right-concatenation of ‘a’ and ‘m’ is the string STR. **Suggested remedy:** implement accordingly (or use alternative equivalent formulation). Rewrite this as follows: “The m data field shall be set to the (unsecured) Payload Field of the MAC Payload if frame security includes providing confidentiality and shall be set to the empty string otherwise. The a data field shall be set to the unique string so that the right-concatenation of the a data field and the m data field is equal to the MAC frame, with the FCS field purged.
3. [CID #30] (TR) Clause 7.3.5.3, p. 151, l. 38-39: This incorrectly delves into internal details of the cryptographic mode of operation CCM\*. The external interface only involves strings a, m, and c, \*not\* an authentication tag or encryption operation (both ‘under the hood’). More importantly, though, the statement is incorrect for incoming frames that were authenticated, but not encrypted: for those frames, the input ‘c’ (which is the authentication tag) of the CCM\* inverse operation is “purged” from the unsecured payload field, i.e., the unsecured payload field results from removing the rightmost substring ‘c’ from the received payload field. **Suggested remedy:** Rewrite this sentence as follows: “The unsecured payload field of the MAC Payload shall be set to the string ‘m’ if frame security includes providing confidentiality and shall be set to the (secured) Payload field of the MAC Payload, with the rightmost substring ‘c’ purged, otherwise.” Note RS: a similar change needs to be made with the incoming frame security procedure (Clause 7.2.3, Step n), since also not entirely correct.

COMMENTS BELOW - NOT SUBMITTED WITH SPONSOR BALLOT RECIRC D05

1. (T) Clause 7.4.1.1, p. 143, l. 24: Replace “MIC)0” by “MIC-0”.**Suggested remedy:** implement accordingly.
2. (T) Clause 7.4.1.2, p. 143, l. 37: Replace “see 7.7.2.4” by “see 7.4.3”. **Suggested remedy:** implement accordingly.
3. (T) Clause 7.4.1.2, p. 143, l. 39: Replace “see 7.7.2.4” by “see 7.4.3”. **Suggested remedy:** implement accordingly.
4. (T) Clause 7.4.3, p. 143, l. 53: Replace “see 7.7.2.2.2” by “see 7.4.1.2”. **Suggested remedy:** implement accordingly.
5. (T) Clause 7.4.3.1, p. 144, l. 33: Replace this sentence by the following sentence: “The Key Source field, when present, indicates the originator of a group key as indicated according to the value speci­fied by the Key Identifier Mode field of the security control field (see ).” **Suggested remedy:** implement accordingly. (Note RS: Thus, resulting in an augmented sentence.)
6. (T) Clause 7.3.2, p. 139, l. 4: Replace “in B2.2” by “in B3.2”. **Suggested remedy**: implement accordingly.
7. (T) Clause 7.3.2, p. 139, l. 18: Replace “see 7.2.2” by “see 7.4”. **Suggested remedy:** implement accordingly.
8. (E) Clause 7.3.3, p. 139, l. 28: Replace “CCM\* decryption and authentication checking process” by “CCM\* decryption and authentication checking transformation”. **Suggested remedy:** Implement accordingly.
9. (E) Clause 7.3.3, p. 139, l. 32: Replace “bit ordering” by “integer and octet ordering”. **Suggested remedy:** implement accordingly.
10. (E) Clause 7.3.3, p. 139, l. 28-36: This text can be deleted entirely, since a repeat of Annex B3.2. **Suggested remedy:** implement accordingly.
11. (TR) Clause 7.2.1, p. 132, Step j), l. 23: Technically, encryption is applied to the string ‘m’ and the authentication tag alike. In particular, even if only authentication is applied, then the encryption transformation of Annex B4.1.3 is still applied, but only over the authentication tag. **Suggested remedy:** Rewrite this so as to be consistent with Clause 7.3 (Security operations). In fact, why not simply remove l. 22-34 (Sub-steps 1-3) entirely, since already spelled out in Clause 7.3.4?
12. (TR) Clause 7.3.4.2, Table 54, p. 140: The terms “secured” (and “unsecured”) are too loosely defined here, since “secured” means “encrypted” here, but generally may include anything resulting from applying security services (including authentication [in which case the MIC field is inadvertently referred to twice in the table]). **Suggested remedy:** Replace “secured” by “encrypted”.
13. (T) Clause 7.3.4.2, Table 54: p. 140: The authentication tag is actually an \*encrypted\* authentication tag (if one looks under the hood of the CCM\* forward operation), no matter whether confidentiality over the outgoing frame is provided or not. **Suggested remedy:** Avoid this altogether by just referring to parameters that show up in the external interface of the CCM\* mode of operation, rather than diving “under the hood”.
14. (TR) Clause 7.3.5.2, Table 55, p. 141:The terms “secured” (and “unsecured”) are too loosely defined here, since “secured” means “encrypted” here, but generally may include anything resulting from applying security services (including authentication [in which case the MIC field is inadvertently referred to twice in the table]). **Suggested remedy:** Replace “secured” by “encrypted”.
15. (T) Clause 7.3.5.2, Table 55: p. 141: The authentication tag is actually an \*encrypted\* authentication tag (if one looks under the hood of the CCM\* forward operation), no matter whether confidentiality over the outgoing frame is provided or not. Moreover, when encryption was not applied to the frame, the “secured” payload field (minus the authentication tag) is actually just the same as the unsecured payload field. **Suggested remedy:** Avoid this altogether by just referring to parameters that show up in the external interface of the CCM\* mode of operation, rather than diving “under the hood”.
16. (TR) Clause 7.2.4, p. 135, Step l), l. 22: Technically, decryption is applied to the string ‘c’, which includes the authentication tag (irrespective of whether encryption was supposedly applied). In particular, even if only authentication was applied, then the encryption transformation of Annex B4.1.3 is still applied, but only over the authentication tag. **Suggested remedy:** Rewrite this so as to be consistent with Clause 7.3 (Security operations). In fact, why not simply remove l. 20-31 (Sub-steps 1-3) entirely, since already spelled out in Clause 7.3.5?
17. (TR) Clause 7.2, p. 131, l. 20: There intended behavior when the *macSecurityEnabled* attribute is set to FALSE is not specified. **Suggested remedy:** Replace “with security when” by “with security only if”.
18. (E) Clause 7.2.1, p. 132, l. 2 (Step e): For consistency of style, one should replace “return with the secured frame and a status of SUCCESS” by “return with a status of SUCCESS”. **Suggested remedy:** Implement accordingly.
19. (T) Clause 7.2.4, p. 134, l. 3-5: The output of the incoming frame security procedure is not in line with what is done elsewhere. As an example, with 7.2.2 and elsewhere, one has something along the following lines: “the outputs from this procedure are a passed or failed status and, if passed, …”. **Suggested remedy:** Use similar language for similar cases (parallelism of style), as suggested.
20. (E) Clause 7.2.4, p. 134, l. 13: should one or shouldn’t one add “as applicable” (as does Clause 7.2.1, p. 131, l. 33)? Not sure what “as applicable” means here, since intent clearly is that all steps are executed (in order) and not just a subset to one’s liking and that one only exits the routine prematurely if an error condition has been met. (Also elsewhere). **Suggested remedy:** Again, consistency of style is key here, as suggested.
21. (TR) Clause 7.2.4, p. 134, l. 10-11: This statement seems more a configuration remark and seems to be redundant. If not redundant, the statement does not seem to be complete, since it does not make similar assumptions on frame counters, short addresses, security level policy settings, etc. – which should also all be configured. **Suggested remedy:** Remove this sentence, make this a Note, or otherwise, and address incompleteness remark illustrated in comment.
22. (E) Clause 7.2.4, p. 134-135, Step h, l. 54-l. 2 (on p.135): To be consistent with style elsewhere, shorten this sentence by stipulating “return with a status of SUCCESS, if <Condition A>, and with a status of <Stat-Code-B> otherwise. **Suggested remedy:** Implement accordingly.