Public Comments on SP 800-131A Rev. 3, Transitioning the Use of Cryptographic Algorithms and Key Lengths

Comment period: October 21, 2024 – December 4, 2024

On October 21, 2024, NIST published NIST SP 800-131A Rev. 3 (Initial Public Draft) Transitioning the Use of Cryptographic Algorithms and Key Lengths. NIST solicited public comments to be sent by email until December 4, 2024. This document provides a compilation of received public comments. Personalized headers and footers have been removed.

LIST OF COMMENTS

1. From: Thomas Pornin (NCC Group), Date: October 22, 2024

In the new SP 800-131Ar3 draft, in section 3 (digital signatures), lines 366-373, it is written that curves such that “224 <= len(n) < 256” are acceptable for generating signatures, but will be deprecated after year 2030, whereas curves such that “len(n) >= 256” will remain acceptable after 2030. Since n is “the order of the base point G” (line 328), a consequence of these statements is that use of Curve25519 in EdDSA (i.e. the very common “Ed25519”) will be deprecated after 2030: in that curve, the whole curve order is slightly above 2^255, but the base point G has an order which is only slightly above 2^252, which means that len(n) = 253.

Since Curve25519 is recommended in SP 800-186, and the new SP 800-131Ar3 draft explicitly follows SP 800-186 (line 362), I suppose that this deprecation side-effect is unintended. I suggest slightly lowering the cutoff length, e.g. into “224 <= len(n) < 250” and “len(n) >= 250”, so that Curve25519 remains formally acceptable after 2030.

2. From: Marcin Fijalkowski, Date: October 23, 2024

A paper was received entitled "An HSM-based EUDI wallet using Split ECDSA (SECDSA) providing verifiable "sole control", written by Eric R. Verheul and dated 13 October 2024.

3. From: Jack Lloyd, Date: October 23. 2024

The current draft SP800-131a states

> Signature verification of EdDSA digital signatures is acceptable using the recommended elliptic curves included in [SP 800-186] where len(n) ≥ 256.

SP 800-186 clearly indicates (correctly) that for Ed25519, n is 253 bits. Likewise Table 3 prohibits EdDSA with strength below 128 bits. Using the convention that the strength of an elliptic curve is approximately half the bit length of the prime order due to Pollard rho, Ed25519 fails this requirement as well, providing at most 126-127 bits of security.

Personally I'd be fine with Ed25519 being prohibited, but I suspect that was not the intent.

4. From: John Mattson (Ericcson), Date: October 29, 2024

Dear NIST,
Thanks for your continuous efforts to produce well-written, user-friendly, and open-access security documents. We welcome NIST's plans to revise SP 800-131A, including the retirement of outdated algorithms like ECB, DSA, SHA-1, and 224-bit hash functions.

- We are not aware of any current or past deployments utilizing 224-bit hash functions.
- We welcome the statement that AES-128 will remain secure for the foreseeable future. This effectively reflects the current state of knowledge, summarized in e.g., [1]. We hope NIST will use the same formulation in other documents.
  Given the existence of questions like Q1 in [2], we think NIST needs to provide an explicit statement also for other algorithms than AES. Readers should understand that all algorithms with a security strength of at least 128 bits, such as SHA-256, SHA3-256, HMAC-SHA-256, KMAC128, and Ascon will remain secure for the foreseeable future. Appendix A is not clear enough.
- We suggest that NIST disallow the use of ECB for all use cases and update NIST specifications where "ECB mode" remains acceptable. For example, one sentence in NIST SP 800-73pt2-5 could be revised to state: "The 16-byte IV SHALL be generated by encrypting the encryption counter with SK_{ENC} using the AES Cipher() function" with a reference to FIPS 197. We do not consider the application of the AES Cipher() function to a single block as a mode of operation.
  We have encountered individuals who mistakenly believe that ECB is safe to use for everything because QUIC and DTLS 1.3 use "ECB". This misconception is highly dangerous, and we fully support NIST's position that using ECB for protecting data constitutes a severe security vulnerability [3].
- The document specifies that encryption using TDEA is disallowed, while decryption is allowed for legacy use. To enhance clarity, we recommend explicitly stating that the use of TDEA for confidentiality protection of data in storage is prohibited, as the encryption may have happened in the past. Data still requiring confidentiality protection must be re-encrypted using AES. Similar considerations apply to stored data with RSA-1024 and SHA-1 signatures, which might need to be re-signed. We think NIST should provide guidance on reprotection of stored data. Reprotection will be necessary also in the transition to quantum-resistant algorithms.
- *"RSA: RSA keys are generated with respect to a modulus n, which is used to determine the security strength that can be provided by a digital signature. The RSA algorithm for digital signatures is specified in [RFC 8017], and guidance for use is provided in FIPS 186."*
  There are several RSA algorithms for digital signatures and FIPS 186-5 provides more than guidance. We suggest:
  "RSA-based Digital Signatures (RSASSA-PKCS1-v1_5 and RSASSA-PSS): RSA keys are generated with respect to a modulus n, which is used to determine the security strength that can be provided by a digital signature. RSASSA-PKCS1-v1_5 and RSASSA-PSS are specified in [RFC 8017], and further requirements are provided in FIPS 186."

- *"ECDSA and EdDSA signature generation providing at least 128 bits of security is acceptable. These signatures shall be generated using elliptic curves and private keys such that len(n) ≥ 256"*
  This is inconsistent and should be changed to make it clear that Ed25519 is acceptable. SP 800-186 correctly states that Edwards25519, with len(n) ≈ 252, offers a security strength of 128 bits. When considering the actual number of low-level operations, Ed25519 provides stronger security than AES-128 against classical computers. One solution would be to remove all mentions of len(n), as it is overly technical and unnecessary, given that SP 800-186 already lists the security strength of all relevant curves.
  - We recommend that NIST allow key agreement using Curve25519 and Curve448. Currently, Curve25519 is used in the vast majority of TLS, DTLS, QUIC, and SSH connections, and this is expected to continue after the transition to quantum-resistant cryptography, as X25519MLKEM768 is anticipated to dominate future implementations. Ericsson is planning to transition as much as possible to Curve25519 and Curve448 during this shift.
  While there is nothing inherently wrong with NIST P-curves or Brainpool, Curve25519 and Curve448 offer superior performance, encoding efficiency, and robustness. A significant issue in consumer and industry products is the lack of public-key validation in some implementations. Given the widespread deployment and superior properties, it would make a lot of sense for NIST to approve Curve25519 and Curve448 for key agreement.

John Preuß Mattsson,
Expert Cryptographic Algorithms and Security Protocols

[1] IETF Statement on Quantum Safe Cryptographic Protocol Inventory
https://datatracker.ietf.org/liaison/1942/
[2] 3GPP Statement on PQC Migration
https://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_118_Hyderabad/docs/S3-244307.zip
[3] NIST, "Announcement of Proposal to Revise Special Publication 800-38A"
https://csrc.nist.gov/news/2022/proposal-to-revise-sp-800-38a

5. From: Andrew Waterhouse (Pacific Research PTY Ltd.), Date: October 29, 2024

I sit on ISO WC-68 SC2 working groups WG13 and WG 11 which, inter alia, are responsible *for ISO 2038 Banking and related financial services — Key wrap using AES key wrap*. This uses a combination of AES-CMAC for authentication, and AES-CBC for encryption using keys derived with an approved KDF. It is a clone of ANSI X9.143/ TR31 but limited to AES usage only.

We have just come across the key wrap prescriptions on P26 of *NIST SP 800-131Ar3 ipd (Initial Public Draft)* which a) seem unclear, and b) if ruling out ISO 2038 key wrap, very problematic for the financial industry and having no obvious security basis. Part of the prescription includes the words "…as well as combinations of an **approved** encryption mode (e.g., AES-CBC) with an **approved** authentication method (e.g., HMAC or a digital signature). However, the table below (Table 11. Approval status of block cipher algorithms used for key wrapping) seems to preclude both AES-CBC and AES-CMAC.

I would greatly appreciate a clarification on this as it has very significant implications for the global banking community.

6. From: John Mattson (Ericcson), Date: October 31, 2024

Here is a list of notes about wording, style, and other non-content issues one of my collegues made when he reviewed 800-131A Rev. 3 (Initial Public Draft).

- Line 496 refers to table 6, but the actual table is numbered 5 (**note**: all table references after this are off by one)
- Line 528 the last letter “d” in “disallowed” is not bold
- Line 568 refers to table 6, should be 7
- Line 579 the text “that provides” is in different font from rest of the text
- Line 619 refers to table 8 as approved **one-step** KDFs, but table 8 contains **two-step** KDFs – the reference should be to table 7
- Line 638 references table 9 as two-step KDM, but it is table 8 that ought to have been referred to
- Line 661 wrong table reference
- Line 662, the table title “Approval status of the algorithms used for a key derivation function (KDF)” does not match what is in text on line 661 “approval status of the PRFs for key derivation” – suggest clarifying as “Approval status of the algorithms based on a PRF and a key-derivation key used for a key derivation function (KDF)”
- Lines 671 and 674 refer to “a transaction” without being explicit about what type of transaction is referred (earlier similar references are explicit on WHAT type of transaction is being discussed); what type of transaction is this? Financial transaction? Probably refers to key derivation or key exchange (as earlier), but as now, I consider this a “loose” term.
- Line 684 wrong table reference
- Line 685, table 10, earlier text (line 683) mentions **approved** hash functions, but table contains “all other hash functions” (does this include unapproved ones too?), suggest changing to “All other approved hash functions”
- Line 703 wrong table reference
- Line 742 wrong table reference
- Line 772 wrong table reference
- Line 796 / table 14 uses “112 ≤ strength < 128”, earlier table 3 uses “≥ 112 but < 128 bits of security strength” and different column title, suggest using only one form across the whole document for consistency

7. From: Neil Madden, Date: November 1, 2024

"The length of *n* (i.e., the domain parameter that specifies the order of the base point *G*) is used to determine the security strength that can be provided by a properly generated key." (lines 327–9)

"The security strength provided by an elliptic curve signature is 1/2 of the length of the domain parameter *n*." (lines 360–1)

"ECDSA and EdDSA signature generation providing at least 128 bits of security is **acceptable**. These signatures **shall** be generated using elliptic curves and private keys such that **len**(*n*) ≥ 256." (lines 371–3)

This requirement is repeated in Table 3, where it states that EdDSA is acceptable for "≥ 128 bits of security strength".

These statements together rule out the use of EdDSA with the edwards25519 curve as specified in FIPS 186-5. This curve only has 126.5 bit security strength according to this definition. I do not believe this is intended, so either the strength requirement should be reduced slightly to >= 126 bits, or else an explicit exception should be carved out to permit edwards25519.

8. From: Michael Williamson (Western Digital), Date: November 4, 2024

The iteration count requirement within 9.3 Key-Derivation in SP 800-132 of SP 800-131A Rev. 3 (Draft) specifies a minimum iteration count of 1000. This supersedes the iteration count requirement specified in 5.2 The Iteration Count (C) of SP 800-132 Recommendation for Password-Based Key Derivation: Part 1: Storage Applications, which states that a minimum iteration of 1000 is recommended, not required.

Is it the intent of NIST to use SP 800-131a to supersede the iteration count requirements in SP 800-132 or does NIST intend to update SP 800-132 to align with the PBKDF2 iteration count requirement in revision 3 of SP 800-131a?

9. From: Ga-Wai Chin (Infineon Technologies), Date: November 25, 2024

Comment on line 715:
Please do not deprecate the use of an approved encryption mode and an approved authentication method for key wrapping. It is widely used in the industry. One example is in the TPM (Trusted Platform Module), which is standardized by the TCG (Trusted Computing Group) and validated under FIPS 140 by multiple vendors.
The TPM uses the combination of AES (CFB) and HMAC to wrap any key material. The current TPM standard does not support use of an authenticated encryption mode (such as KW/P, CCM and GCM) instead. Deprecating the use of this algorithm will create a negative perception towards the security of a TPM for our customers.

Should the final SP 800-131Ar3 deprecate the general use of an approved encryption mode and an approved authentication method for key wrapping, then we need a guidance to be published by NIST - at the same time the final SP 800-131Ar3 is published - that approves the combination of AES CFB and HMAC for key wrapping as used by the TPM.

10. From: Graham Costa (Thales), Date: November 26, 2024

We appreciate the opportunity to comment on your draft special publication. We've collated our comments from Thales DIS and where these can be found below.

This is an important document to the cryptographic community and where we view this as an important opportunity to work with NIST to minimize the impact transitions outlined in this document may have on consumers of cryptography not just within the US federal government but in many cases globally.

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
Thales_1	section 2.2, lines 291.	Technical	Whilst we don't object to disallowing ECB, vendors need a transition period associated with this change (similar to removing 224-bit hash). Without a transition period, the withdrawal of ECB mode would happen on immediate publication of SP 800-131Ar3. Whilst we endorse that there are few reasons to continue to support ECB in general cryptographic libraries, it is part of many cryptographic modules today based on it's long-term inclusion in SP 800-38A and where for FIPS 140-3 certified modules at all levels, removing the algorithm from general purpose use will require firmware changes. In particular, at Level 3, modules will need to block any attempts to use ECB mode with approved services,	We'd recommend deprecating ECB mode until Dec 31, 2030 and then disallowing beyond this. This would be consistent with 224-bit hash algorithms. We'd also strongly encourage NIST CTG to update SP 800-38A formally first. This should identify use-case for ECB and only when that standard is updated, start to transition away from ECB. Ultimately if used appropriately, ECB is secure. As such, although it is recommended to move away from it as a standalone mode acknowledging the potential risks of inappropriate use, disallowing it at no notice and without giving industry time to plan for the transition is inappropriate, unnecessary and inconsistent with all prior transitions triggered by SP 800-131A.

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
			and at all levels, approved security service indicators will need updated to correctly reflect ECB mode as a non-approved algorithm.	
Thales_2	section 1.2.2, lines 181-186.	Technical	Legacy use is defined in the document as: *"The algorithm or key length may only be used to process already protected information (e.g., decrypt ciphertext data or verify a digital signature)."* This is problematic in that this has been interpreted by CMVP as meaning that all data used with a algorithm classed as 'legacy' needs to be provably have been protected by the algorithm ahead of date the algorithm transitioned to legacy. Our original reading of 'legacy' was that NIST had pragmatically created this category to allow older cryptographic systems to interact with (or support migration to) new systems. In the case of moving long-term keys (which may be valid for up to 30 years) this could involve encrypting these in an existing HSM that exclusively supports legacy algorithms for these keys to them be imported into a new module that supports the latest	Add the following footnote linked to 'already protected information': "already protected information may have either: • had protection applied prior to target algorithms transition to legacy; or • has had protection applied by an older module to support interoperability or migration of keys to a newer cryptographic module. In this second example, objects would be protected after the transition of an algorithm to legacy but where this occurs outside the boundary of a module compliant to all requirements of SP 800-131Ar3 and is permitted exclusively for module interoperability during transition periods between different cryptographic algorithms ahead of them becoming

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
			algorithms but also legacy encryption options for import. The problem with the current definition of 'legacy' is the different readings of 'already protected information'. In our case, we want to read this as it being OK to decrypt key objects passed into a current HSM from an older HSM being retired. As mentioned above, CMVPs first read of above is to tie 'already protected' to a timeline linked to the when the algorithm transitioned to legacy. This doesn't work for support crypto estate migration to newer equipment where encryption using the legacy algorithm would need to occur today but where the current generation module would exclusively support decryption using the legacy algorithm.	disallowed, and separately also for key migration."
Thales_3	section 10, line 705.	Technical	Section 10, deprecates the use of independent encryption and authentication for key wrapping. Given this is permitted by SP 800-38F and where there are no security risks associated with this method if implemented correctly, we don't	Update Table 11, to make 'key wrapping using separate encryption and authentication processes' approved. Work with the CMVP program to create a short IG linked to appropriate use of encryption paired with authentication techniques for key transport as part of FIPS 140-3, IG D.G.

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
			understand why this is suddenly being deprecated. The ability to support key wrapping with independent encryption and authentication is important to a number of protocols and where prior to deprecating this method with immediate effect, we strongly feel NIST should discuss this proposed change with industry first through a draft update to SP 800-38F. A large number of modules will support this arrangement for key transport and where at minimum there should be a extended transition period ahead of any algorithm being transitioned to deprecated or disallowed. This has an immediate effect of vendors and in many cases could lead to a breaking change for existing protocols where transitions and updates need to be carefully managed and staged with customers. Where the draft standard identifies the need for deprecation as being the need for additional guidance ("The use of an approved encryption mode and an approved authentication method for key wrapping is deprecated until additional	

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
			guidance is provided for using these combinations securely."), there are many algorithms or modes such as ML-KEM and GCM where additional guidance is broadly acknowledged by the cryptographic community as being needed but where NIST hasn't on a temporary basis 'deprecated' these algorithms in response.	
Thales_4	section 1.2.3, lines 233-237.	Technical	For the avoidance of doubt, should this first bullet also mention key generation and message authentication codes? In some cases for these function the 112-bits security strength is based on the length of key used and not necessarily exclusively on the hash function used. In general, the first bullet mentions exclusively hashing and block ciphers. The second mentions explicitly signatures. There are other categories of function listed in SP 800-131Ar3.ipd that would seem to fall into a gap between the two current bullets.	Expand bullets one and two to make it clear where all functions by section of the current standard are now considered to fall E.g. is a truncated HMAC that may offer less than 128-bit security permitted after Dec 31, 2030.
Thales_5	section 6, line 540.	Technical	This standard references use of curves K-283, K-409, K-571, B-283, B-409, B-571' with both FIPS PUB 186 (ECDSA	Update the first bullet to remove the binary curves as acceptable.

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
			and EdDSA) and SP 800-56 (ECDH). Since SP 800-186 deprecates use of the binary curves 'K-283, K-409, K-571, B-283, B-409, B-571' should the corresponding entries in Table 5 now also be listed as deprecated when using these curves? This will result in the need for a duplicate entry in Table 5 for SP 800-56A DH and MQV schemes using finite fields and elliptic curves that currently are listed as acceptable through 2030 with no mention of deprecated curves.	Add a new bullet to cover use of deprecated curves with key agreement: "- Key-agreement transactions providing at least 128 bits of security strength using the following elliptic curves are **deprecated**: o K-283, K-409, K-571, B-283, B-409, B-571, as specified in [SP 800-186]);"
Thales_7	section 9, lines 610 and 611.	Technical	This section currently explicitly links use of SP 800-56C based KDF to SP 800-56A and SP 800-56B. In addition to this, following comments from the community, SP 800-56C KDF are also explicitly referenced and permitted for ML-KEM (FIPS 203). As such, for the avoidance of doubt, can explicit reference to FIPS 203 also be made from sections referencing use of SP 800-56C.	Update: "[SP 800-56C] provides key-derivation methods (KDMs) for key-establishment schemes in [SP 800-56A] and [SP 800-56B]" to "[SP 800-56C] provides key-derivation methods (KDMs) for key-establishment schemes in [SP 800-56A], [SP 800-56B] alongside the key encapsulation mechanism (KEM) in [FIPS 203]."
Thales_8	Notes to Reviewers, lines 29-30.	Technical	NIST posed the following question in the introduction section of the initial public draft:	N/A.

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
			"Question: Does the retirement date of December 31, 2030, for the 224-bit hash functions pose an unacceptable burden on implementers or users" In response to this question, Thales has been unable to find any challenges linked to disallowing 224-bit hash in this time-frame. We have considered the changes in the context of software libraries, HSM and independently for SmartCard, Secure-IC and products derived from them. This length of hash isn't critical to any existing identified protocols and where we do have libraries that support these hash variants for general use, 5 years is an adequate time-frame for us to transition support for these algorithms out of any configurations tied to strict implementation of NIST approved cryptography.	
Thales_9	various.	technical	Since standardization of ASCON is in progress with the initial public draft now out for review, we suggest adding mention of it to appropriate sections in SP 800-131Ar3. In particular, it would be particularly helpful to identify all sections where ASCON will ultimately be included. e.g.	Add ASCON (SP 800-232) to the relevant tables but include a footnote confirming that the standard is still in development consistent with how FIPS PUB 186-5 was represented initially in SP 800-131Ar2.

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
			Will it be allowed for key wrapping? How does NIST expect the hashing and XOF functions to be used i.e. Will they be able to be used with the various signature standards etc. Including forewarning of the positioning for ASCON once would be consistent with how SP800-131ar2, included references to FIPS PUB 186-5 long before it's formal publication.	
Thales_10	section 10, lines 694-704.	technical	In addition to options set out in section 10 for key wrapping, FIPS 140-3, IG D.G includes the following statement: *"Allowed methods for key transport in an approved mode.* ● *...* ● *A key unwrapping using any approved mode of AES or two-key or three-key Triple-DES. Key wrapping is not allowed if the algorithm does not meet the requirements of SP 800-38F. "* To avoid this being missed by users, can this be added in a footnote in section 10. The use-case for this inclusion is to support decryption of keys imported to newer generation of cryptographic	Include relevant statements from FIPS 140-3, IG D.G in section 10 to avoid these allowances being missed and to provide a concrete foundation to developers relying on this option for key import from retired cryptographic modules developed prior to SP 800-38F being published.

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
			module (as part of key migration on retirement of older modules).	
Thales_12	section 1.2.2, line 178.	Technical	Since selection of an appropriate curve can lead to a given algorithm being considered allowed or deprecated (SP 80-186), we'd suggest curve should explicitly be called out in this sentence.	Update: "deprecated algorithm or key length" to "deprecated algorithm, key length or curve".
Thales_13	section 1.1.	technical	The release version of SP 800-131Ar3, should add reference to NIST IR 8547,*'Transition to Post-Quantum Cryptography (PQC) Standards'* to record that this document will contain, once published, the transition timeline for PQC. Since we know there is a time-line and that it's broadly falling in the same time-frame as some of the transitions already outlined in SP 800-131Ar3, the document should include acknowledgement of the transition document and its status at time of publication of SP 800-131Ar3.	Add a paragraph to section 1.1. to acknowledge that NIST is working on a broader strategy to PQC compliant cryptography by timelines set out in National Security Memorandum 11, (NSM-11). Example paragraph: "NIST is concurrently working on a transition strategy to post-quantum secure algorithms as part of addressing threats posed by cryptographically relevant quantum computers. This strategy is being developed at this time in NIST IR 8547, 'Transition to Post-Quantum Cryptography Standards' and where this looks to meet targets set out in National Security Memorandum 11 (NSM-11). An initial public draft of this document is currently open for review."

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
				or similar pending what-ever comes out of the initial public review period for NIST IR 8547 and time-line to publication of SP 800-131Ar3.
Thales_14	section 6.	Editorial	Since the references section includes both a '[SP800-56Ar2]' alongside '[SP 800-56Ar3]' bookmark, we suggest references throughout the document to '[SP 800-56]' are updated to '[800-56Ar3]' to make it clear that only this version is accepted and in particular where this version adds explicit requirements linked to key validation not present in earlier revisions.	Update references to '[SP 800-56A]' bookmark to '[SP 800-56r3]'. Caveat: It will still be necessary in a small number of places to explicitly reference SP 800-56Ar2 linked to 112 bit security strength algorithm options.
Thales_15	section 1.2.1, lines 159, 305-206, 362.	Editorial	Since the FIPS 140 Implementation Guidance document could be re-ordered and renumbered and/or in the future we may have a new implementation guidance document for FIPS 140-4 that contains different IG numbering to the FIPS 140-3 IG we suggest including the full title of any explicitly referenced IG. In this case, this would also clarify that IG D.B only covers security strengths of SSP establishment methods and not 'each algorithm' as	Example update: "FIPS 140 Implementation Guide [FIPS 140 IG] Annex D.B." to "FIPS 140 Implementation Guide [FIPS 140 IG] Annex D.B, Strength of SSP Establishment Methods." Other direct references to the FIPS 140-3 IG exist and should independently be updated for IG C.A, C,H, D.B, D.J, D.K, D.O and 9.3.A

Comment Number	Section / Line Number	Comment Type	Comment	Proposed Resolution
			implied by the sentence preceding the reference. Expanding the title of FIPS 140-3, IG C.H and C.A exclusively serves to future proof the reference as the FIPS IG evolves.	
Thales_16	various sections, lines 130-133 and 142-149 (examples and not exclusive references).	Editorial	This document uses inline numbered lists in a number of places and where these are hard parse and where this style is inconsistent with other NIST SP. An example is shown below: "This guidance is intended to 1) encourage the specification and implementation of appropriate key-management procedures, 2) use algorithms that adequately protect sensitive information, and 3) plan for possible changes in the use of cryptographic algorithms, including any migration to different algorithms and key lengths."	We propose items in each numbered list are put on separate lines to aid the reader and to improve the ability to cross-reference the lists. "This guidance is intended to: 1. encourage the specification and implementation of appropriate key-management procedures; 2. use algorithms that adequately protect sensitive information; and 3. plan for possible changes in the use of cryptographic algorithms, including any migration to different algorithms and key lengths."
Thales_17	section 6, various.	Editorial (minor)	Line 528. Missing bold final 'd' on end of '**disallowe**d'.	<As per comment column.>

11. From: Ignacio Diéguez (Entrust), Date: November 29, 2024

#	Type	Line#	Comment (Include rationale for comment)	Suggested change
s6	Clarification	531	This section does not mention B-233 or K-233 as either acceptable, deprecated or disallowed.	Make explicit the status of B-233 and K-233.
	Te	29	*Question: Does the retirement date of December 31, 2030, for the 224-bit hash functions pose an unacceptable burden on implementers or users?* From our point of view it doesn't pose an unacceptable burden. SHA-256 is a more natural migration path from SHA-1.	
	Ed	621	Typo in the table column heading "Crypto.Primitive".	Remove the "." between "Crypto" and "Primitive".
	Ge		We agree with the disallowance of the ECB mode, but it should have a timeline and transition period, to allow vendors to plan and mitigate the impact. For example, already planned CMVP submissions will be disrupted if the disallowance has immediate effect.	Add a timeline for disallowance of ECB mode, instead of having immediate effect.
	Ge	715	Entrust is very worried with the deprecation of key wrapping using separate approved encryption and authentication processes and we look forward to seeing them be listed back as "Acceptable" once the additional guidance for using these combinations securely is published. Key wrapping using separate approved encryption and MAC mechanism is widely used securely in HSMs for storing key material and associated attributes securely, in an encrypt-then-MAC fashion. These may include proprietary schemes, or standardized schemes such as TR-31 key block format in X9.24-1, now widely used within the payment industry, and Global Platform secure channel protocols SCP03/SCP04 widely used to communicate and provision smartcards, also make uses of	Consider removing this deprecation until additional guidance is published and industry has a chance to provide feedback. Once the final guidance is published, those mechanisms using separate encryption and MAC following the guidance

#	Type	Line#	Comment (Include rationale for comment)	Suggested change
			these schemes, e.g. refer to section 6.10 Sensitive Data Encryption and Decryption in https://globalplatform.org/wp-content/uploads/2022/07/GPC_2.3_K_SCP04_v0.0.0.24.pdf In these scenarios, the wrapping and unwrapping entities are known and are evaluated with FIPS 140 to diligently verify the MAC before proceeding with the key material decryption.	should be categorised as "Acceptable".
	Te	765	*The use of TupleHash and ParallelHash is acceptable for the purposes specified in [SP 800-185].* It is not clear where in SP 800-185 it is specified the acceptable uses of TupleHash and ParallelHash.	Please clarify.
	Ed	776	Typo: empty line 776.	Remove.

12. From: Gil Bernabeu (Global Platform), Date: December 4, 2024

1. Overall Description

GlobalPlatform would like to make the following comment to the current ipd.

GlobalPlatform technical comment on Section 10, Line 705, Table 11. "Approval status of block cipher algorithms used for key wrapping"

This Table deprecates the use of independent encryption and authentication for key wrapping, without any additional guidance, and without any transition period before being transitioned to deprecated or disallowed.

1. GlobalPlatform Secure Channel Protocol 03 (SCP03) uses AES-CMAC for the derivation of a session key. GlobalPlatform interprets this as acceptable according to the new draft version of SP 800-131A.

2. As soon as the SCP03 session is running, a key can be sent to the secure element (SE), encrypted using AES-CBC and authenticated using AES-CMAC (with different keys for CBC and MAC).

• _GlobalPlatform interprets this as a key wrapping, making SCP03 incompatible with the new draft version of SP 800-131A. Is this interpretation correct?

• _Given this is permitted by SP 800-38F and where there are no security risks associated with this method if implemented correctly, GlobalPlatform does not understand why this is suddenly being deprecated.

2. Actions:

GlobalPlatform suggests updating Table 11, to make 'key wrapping using separate encryption and authentication processes' approved.

13. From: Bryan Queen (NSA), Date: December 4, 2024

Page #	Starting Line #	Ending Line #	Section #	Comment/Rationale	Proposed Change
2	167	165	1.2.1	Editorial: The explanation of "classical security strength" (and the document's implicit use of "security strength" to mean "classical security strength") is important enough to appear in the main body of the text.	Elevate what is now in Footnote 2 into the text of section 1.2.1. The beginning of the section (before line 156) would be an appropriate spot -- before the term "classical security" is first used (currently without explanation).
2	169	171	1.2.2	Editorial: To say that "… approval status for an algorithm will also depend on … any domain parameters …" is a bit vague, and may not cover enough ground: What about things like the choice of MAC function, or the choice of Hash function used by HMAC? Maybe just "parameters" would be better. (That would be consistent with the repeated use of "parameter sets" in the definitions that follow.)	Rephrase the sentence: "Often, the approval status for an algorithm will also depend on the length and/or strength of its key(s), the choice of implementation-related parameters (if any), and the mode or manner in which the algorithm is used."
3	207	208	1.2.2	Grammar: "The use of algorithms and key lengths/strengths . . . involve some risk that increases over time." This is incorrect. Replace "involve" by "involves" in that sentence. Also the terms "deprecated" and "legacy use" in that sentence should appear in quotes. Finally, the applicability of those terms should be expanded to	Correction: "Reliance on any algorithms, key lengths/strengths, parameter sets, and/or modes/manners of use to which the terms "deprecated" or "legacy use" have been applied involves some measure of risk, which increases over time."

				parameter sets, modes of operation, etc. (There are other problems,, but it's easier to just suggest another sentence.)	
3	213	219	1.2.2	This paragraph difficult to parse. (It may be difficult for readers to unravel.) That is particularly true of this sample sentence: "When acceptable or deprecated is used as the status for applying protection, acceptable is used for processing already protected information."	**General suggestion:** Rewrite the paragraph. For the particular sentence, I think what was intended was something like this: When “acceptable” or “deprecated” is used as the status for applying protection, “acceptable” is used as the status for processing already protected information. **Suggestion:** Lose the "used as" expressions and emphasize "approval status." **Try:** If “acceptable” or “deprecated” is the approval status for applying protection, then “acceptable” is the approval status for processing already protected information. **Full rewrite (first draft):** "This document uses the terms "acceptable," "deprecated," and "disallowed" to indicate the approval status of various algorithms, key lengths/strengths, parameter sets, etc. with respect to their use in applying cryptographic protection to data (e.g., encrypting data, or generating a MAC tag or digital signature). The terms "acceptable" and "legacy use" are used to indicate the approval status of various algorithms, key lengths/strengths, parameter sets, etc. when they are considered for use in processing

					already protected information (e.g., decrypting ciphertext, or verifying a MAC tag or digital signature). If "acceptable" or "deprecated" is the indicated status with respect to applying cryptographic protection, then "acceptable" is the status that applies with respect to the processing of already protected information. If "disallowed" is the indicated status with respect to applying cryptographic protection, then "legacy use" is the status that applies with respect to the processing of already protected information." **[Is that what was intended?]**
3	213	219	1.2.2	There is still risk involved in say verifying a signature or integrity of ciphertext resulting from a deprecated algorithm. The status of “acceptable” seems mis-placed as it indicates no risk on the verifying party.	Consider changing status, perhaps “deprecated” can be re-used.
3	222	224	1.2.3	Rewrite sentence for clarity.	Suggestion: "In particular, the NIST-approved digital-signature schemes based on elliptic-curve cryptography or RSA and the NIST-approved key-establishment schemes using Diffie-Hellman, MQV, or RSA will all need to be

					replaced by secure quantum-resistant counterparts."
4	229	232	1.2.3	Rewrite sentences for clarity**. NOTE:** The 112 and 128 represent old and new MINIMUM security-strength targets. The fact that you're raising the floor (not the ceiling) should be stated at least once.	Suggestion: "For several years, NIST's plan has been to transition from a minimum targeted security strength of 112 bits to a minimum targeted security strength of 128 bits on January 1, 2031. However, since quantum-resistant digital-signature and key-encapsulation mechanisms are now standardized, this revision of SP 800-131A is modifying the transition schedule as follows:"
4	238	239	1.2.3	You must hyphenate "digital signature" when talking about a "digital-signature mechanism."	Suggestion: "Deprecate the use of the 112-bit security strength for the classical digital-signature and key-establishment mechanisms"
4	240	248	1.2.3	Rewrite sentences for clarity.	Suggestion: "Instead of a two-step transition (first moving from a minimum 112-bit security strength to a minimum 128-bit security strength for classical algorithms and later moving from classical algorithms to the approved quantum-resistant algorithms), this revision is proposing a one-step approach whereby the quantum-resistant algorithms are implemented and available as soon as feasible. Currently, a classical digital-signature or key-establishment algorithm targeting a 112-bit security strength does not appear to be in imminent danger of

					becoming insecure, so this approach should allow an orderly transition to quantum-resistant algorithms without unnecessary effort by the cryptographic community."
4	249	250	1.2.3	Rewrite sentence(s) for clarity.	Suggestion: "NIST is developing a schedule for transitioning to the quantum-resistant algorithms. (This is discussed in Sec. 3, 6, and 7.)"
4	251	252	1.2.3	Rewrite sentence for clarity.	Suggestion: "If attacks against digital-signature and/or key-establishment schemes targeting a 112-bit security strength become viable, a transition to the 128-bit security-strength minimum will be required."
5	277	277	2.1	Wrong conjunction. AES can be used with 128-, 192-, **or** 256-bit keys	Rewrite: "Encryption and decryption using AES with either a 128-bit, 192-bit, or 256-bit symmetric key is acceptable."
6	291	291	2.2	Rewrite table entries for clarity.	Under "Status" for SP 800-3G FF1 add "for" to parenthetical remarks: "Acceptable (for domain size of at least one million)" and "Disallowed (for domain size of less than one million)"
6	296	296	2.2	Remove extra space from "(IR)"	Replace "(IR)" by "(IR)" to get: "However, NIST Internal Report (IR) 8459 discusses . . ."
7	308	308	2.2	Missing boldface on "acceptable" for FF1 mode.	"**FF1:** The FF1 mode is **acceptable** when used as specified…"
8	327	329	3	Instead of "length," use "bit length" when describing the parameter n.	Suggested rewrite: "The bit length of the domain parameter n (which specifies the order of the base point G) is used to

					determine the security strength that can be provided by a properly generated key."
8	340	346	3	Nit-picking: I think that it's considered better to completely parenthesize the item numbers used in an in-line numbered list.	Suggestion: "The security strength provided by a digital-signature generation process is no greater than the minimum of (1) the security strength that the digital-signature algorithm can support with a given parameter set (including the length of the key) and (2) the security strength supported by the cryptographic hash method that is used."
10	360	360	3	The guidance following "Signature generation" applies to both the generation and verification of signatures (which use the same EC arithmetic). Pull that out as a separate paragraph placed before the individual bullets for signature generation and signature verification.	**Slightly altered wording (in boldface):** "The security strength provided by an elliptic curve signature is 1/2 of the **bit** length of the domain parameter n. Recommended and deprecated elliptic curves for digital signature generation **and verification** are provided in [SP 800-186]. Elliptic curves that meet the security strength requirements are also allowed when they satisfy the requirements of FIPS 140 Implementation Guide [FIPS 140 IG], Annex C.A."
10	360	361	3	(Again!) Instead of "length," use "bit length" when describing the parameter *n.*	Suggested rewording: "The security strength provided by an elliptic-curve signature is 1/2 of the bit length of the domain parameter *n*."
10	367	368	3	The first time it's used, you might remind the readers that "**len**(n)" denotes the bit length of n. (You don't have to do it again in the following bullets.)	Suggested rewording: "For these curves, 224 ≤ **len**(*n*) < 256, where **len**(*n*) is the bit length of *n*."

10	365	385	3	Please clarify the situation regarding the use of Elliptic curves that satisfy the requirements of FIPS 140 Implementation Guide [FIPS 140 IG], Annex C.A. They are said to be "allowed" for signature generation (without explicitly saying whether they are "acceptable" or "deprecated"), but are unconditionally classified as for "legacy use" in signature verification. That is inconsistent with your (unclear) explanation of the use of terminology on page 3, lines 213-219. A "legacy use" status for signature verification implies that the curves must be "disallowed" for signature generation.	Clearly state the status of the use of Elliptic curves that satisfy the requirements of FIPS 140 Implementation Guide [FIPS 140 IG], Annex C.A. for both signature generation and verification – in a way that is consistent with the discussion of "status assignments" that appears in the last paragraph of Section 1.2.2.
11	387	391	3	(Again!) The guidance following "Signature generation" applies to both the generation and verification of signatures (which use the same RSA parameters). Pull that out as a separate paragraph placed before the individual bullets for signature generation and signature verification.	**Slightly altered wording (in boldface):** " The security strength provided by an RSA signature **scheme** depends on the bit length of the modulus *n*. The security strength associated with several values of **len**(*n*) is provided in [SP 800-57]. The security strength associated with other values of **len**(*n*) may be estimated using the formula in FIPS 140 Implementation Guide [FIPS140_IG], Annex D.B."
13	424	424	4	I don't see how this section is directly applicable to the topic of the document. What does it really tell the reader about transitioning to new algorithms/security-level requirements? It mentions that SP 800-133 cites the required foundation of approved RBGs undergirding everything else – but the only listing two approaches to the generation of symmetric keys is a terribly short summary of SP 800-133 (which references/describes more specific ways to generate various types of keys). I think that there's a reason that this section did not appear in the previous version of this document.	Rethink this section. Most (if not all) of the public-key algorithm specifications include descriptions of (public/private) key generation. Are any of those methods affected by transition? If so, THAT would be useful to point out (perhaps in the sections directly dealing with the affected algorithms). In addition to the two categories of symmetric-key generation that are currently listed here, there are various public-key-based key-establishment methods for obtaining keying material. Aren't those worthy of mention under the heading of "Cryptographic Key Generation"? Of course, those methods have dedicated

					sections in this document (as do RBGs and key-derivation techniques). Aren't all of these covered by the simple guidance that "methods for determining keys are **acceptable** when consistent with the requirements of the application for which the keys will be used." Maybe that's all you needed to say (somewhere else).
14	446	447	5.1	Replace "include" by "employ"	New wording: " . . . which are specified to employ either a hash function or a block cipher (e.g., AES) as cryptographic primitives."
14	461	465	5.2	What's missing is a statement concerning what is acceptable (or not).	Guess: "Validated Entropy sources that comply with the [SP 800-90B] design and testing guidance are **acceptable**."
16	489	491	6	Change "key" to "keys" at the end of the sentence.	Result: "The security strength that can be provided by the algorithm depends on the length of *p*, the length of *q*, and the proper generation of the domain parameters and the **keys**."
16	493	495	6	Replace "The length" by "1/2 of the bit length" in this sentence.	Result: "**1/2 of the bit length** of *n* (i.e., the order of the base point *G*) is used to determine the security strength that can be provided by a properly generated curve."
16	496	496	6	Table 5 is erroneously called Table 6.	Replace "Table 6" by "Table 5" in this sentence.

16	502	502	6	Trying to decipher Table 5. Alternatives (1): Move Table 5 so that it doesn't look like there is a separate row with a blank entry in the first row at the bottom of page 16. That raises an issue though, since there are no domain parameter(-size) sets in the current version (rev 3) of SP 800-56A that are targeted at less than 112 bits of security. Nor were than any such parameter-size sets in rev 2 of SP 800-56A. You have to go back more than a decade to find such domain parameters used. These were disallowed in the previous version of 131A. What does the new "legacy use" status mean?	As in the previous version of SP 800-131A, the Table should show the status “disallow” for the use (initiation) of key-agreement schemes that offer less than 112 bits of security (and also disallow schemes that are not NIST-approved).
16	502	502	6	Trying to decipher Table 5. Alternative (2): Fill in the blank table entry in the top right corner of Table 5 on p. 16 (treating it as part of a separate row) Question: Why wouldn't this row be subsumed by the last row of the table, since no current or recent SP 800-56A (rev3 or rev 2) KA-scheme/parameter-set pair targets less than 112 bits of security. Maybe you should strictly limit the table entries to giving the approval status w/r/t initiating (new) key-agreement transactions. Then, in the text give an explanation of what “legacy use” would mean (since you are not using the phrase in the way it was described earlier in the document). Repeating a previous transaction’s computations is not in the same category of such cryptographically complementary behavior as decrypting ciphertext, verifying a MAC tag, or verifying a signature.	I guess these schemes would be “Formerly NIST-approved/allowed KA scheme/parameter set pairs”? (You would have to be looking back, to those found in 56A rev2, prior to April 2018, and only considering the specific FA/EA-type parameter sizes.) These sorts of schemes were listed as disallowed in the corresponding Table of the previous version of 131A. The status shown in the Table has to distinguish between engaging in new KA transactions (disallowed in this case) and "re-enacting" previous transactions (a legacy use?). Initiating a KA transaction that establishes keying material supports “applying cryptographic protection,” so the status choices should be limited to “acceptable,” “deprecated,” or “disallowed.” That’s the status

					that the table should show. The status (if any) applied to re-computing the keying material at some later date should be forced by that assignment, and so be either “acceptable” or “legacy use.” (I question the value of those assignments. The re-computation of the keying material isn’t what’s of interest, it’s the continued reliance on that keying material.) If legacy use is referring to use of cryptographic keying material derived from a disallowed KA-scheme, then please state so.
17	508	509	6	Be more specific. Also replace “with len(p) = 1024 or len(q) = 160” by “with len(p) = 1024 and len(q) = 160.” That’s what parameter-size set FA required.	Rewrite: “The initiation of an [SP 800-56Ar1]-compliant FFC key-agreement transaction using parameter-size set FA (with len(p) = 1024 and len(q) = 160) is **disallowed**, but the (re-) processing of information obtained from such a transaction is allowed for **legacy use** to re-compute keying material originally produced during a previous SP 800-56Ar1]-compliant FFC key-agreement transaction.”
17	508	509	6	As mentioned in comments to table 5, unclear what is meant by legacy use in this context.	If legacy use is referring to use of cryptographic keying material derived from a disallowed KA-scheme, then please state so.

18	548	549	6	Far too lenient. Why would you want to allow re-computation of key-material that was produced using a potentially awful (disallowed) KA scheme? (There's more to security than the choice of domain parameters.)	Maybe you should require that both the KA scheme and the domain parameters were formerly acceptable in NIST's eyes.
19	557	558	7	**NOTE:** SP 800-56A also describes a "general hybrid method" of key transport (in Section 4.3).	
19	568	568	7	Table 6 is erroneously called Table 7.	Replace "Table 7" by "Table 6" in this sentence.
19	571	571	7	Although you (tried to) explain how approval status assignments work for algorithms – in terms of the original application of protection vs. processing something to which protection has been applied – you never tried to translate that to KE schemes. That makes (Table 5 and) Table 6 entries harder to interpret. Correct that omission, and rethink the status values shown in (Table 5 and) Table 6.	**Initiating a KE transaction** that establishes keying material supports "applying cryptographic protection," so the status choices associated to using a KE scheme should be limited to "**acceptable**," "**deprecated**," or "**disallowed**." **That's the status that Table 6 should show for KE schemes.** The status (if any) applied to re-computing the keying material at some later date (by re-processing transaction data) can be derived from that initiation-related assignment (analogous to the way it was explained in Section 1.2.2): either "acceptable" or "legacy use." (I question the value of those assignments, because re-computation of the keying material isn't what's of interest, it's the continued reliance on that keying material.)
19	571	571	7	The top and bottom entries in the rightmost column of Table 6 should be "disallowed." That applies to initiation of KE schemes, which is the more important scenario,	Replace both occurrences of "legacy use" in Table 6 by "disallowed." (See previous comment.)
19	577	578	7	You've copied (mis)information about 56A schemes into this discussion of 56B schemes.	Suggested Replacement: "The (re-)processing of information obtained from a formerly acceptable RSA-based key-establishment transaction is

					allowed as **legacy use** when **len**(n) = 1024. See the IFC parameters in the August 2009 version of [SP 800-56B]."
19	577	578	7	As covered in comments on table 6, what is meant by legacy use is unclear in this context.	If legacy use is referring to use of cryptographic keying material derived from a disallowed KE scheme, then please state so. Same applies to lines 591-592.

22	605	608	9	The false impression is given that all that's going to be considered is the transformation of a pre-existing key into more keying material. The brief description of the use/functioning of key-derivation methods (KDMs) omits an important scenario. (Which is ironic, since it's the first one discussed in this section). You discuss pre-established KDKs (only), but that begs the question of where they come from. All of the KDMs in SP 800-56C derive keying material from a more general primary source – a shared secret Z that is produced during the course of a KE transaction. That Z is not a KDK, nor are the passwords/pass phrases used in SP 800-132.	Rewrite the section to point out a few more things: (1) SP 800-108 rev 1 specifies approved key-derivation functions (KDFs) that can be used (only) to transform a pre-existing KDK into a bit string that can be parsed into one or more keys and/or other secret bit strings; (2) SP 800-56C rev 2 specifies approved KDMs (one-step and two-step methods) that can be used (only) to transform a shared secret Z – produced during the execution of an approved key-agreement scheme – into a bit string that can be parsed into one or more keys and/or other secret bit strings; (3) SP 800-132 specifies methods for deriving keys from passwords or passphrases; (4) SP 800-135 rev 1 provides additional application-specific KDMs; and (5) SP 800-133 rev 2 specifies even more methods for generating keys of (almost) all types.

22	612	615	9.1	Before talking about what can be employed for "legacy use," take the time to (re)define what the approval status labels mean for a KDM – "**acceptable**," "**deprecated**," or "**disallowed**" would likely apply to the initial key-derivation computation (in an approved/deprecated/disallowed context), while "acceptable" or "legacy use" could (I suppose) be labels applied to repeating a previously performed key-derivation computation. (Again, I'd be more worried about the use of the keying material output by a disallowed KDM than the fact that it was being re-computed.)	Explain what "**acceptable**," "**deprecated**," and "**disallowed**" mean in this section of the document, and then explain "**legacy use.**" (Key derivation was not included in the discussion of Section 1.2.2; explain the equivalents/applicability of "applying cryptographic protection" and "processing protected data" in the context of key derivation.
22	612	614	9.1	Clarify the statement "When a key-derivation method is allowed for legacy use, other conditions specified in Sec. 6 and 7 for the key-establishment schemes also apply."	TRY: "If a key-derivation method is allowed for legacy use, its use is further limited to scenarios involving key-establishment schemes and domain parameters whose approval status is either "acceptable" or "legacy use," as specified in Section 6 or Section 7."
22	619	620	9.1.1	Table 7 is erroneously called Table 8.	Replace "Table 8" by "Table 7" in this sentence.
22	621	621	9.1.1	Change the rightmost entry of the first row to reflect the first-time use status. Legacy use can be described in the text below the table.	New Table Entry: Deprecated through 2030; **Disallowed** after 2030
22	625	626	9.1.1	Conditions not specific enough.	Replacement: "The use of these hash functions for one-step key derivation during an otherwise **acceptable** or **deprecated** key-establishment transaction (see Sections 6 and 7) is **deprecated** through December 31, 2030 and **disallowed** after 2030."

22	627	629	9.1.1	Conditions not specific enough. In particular, why allow computations to be repeated if they were part of a transaction that was disallowed in the first place?	Try: "After 2030, the use of these hash functions is allowed for **legacy use** to (re-)derive keying material using the information from a key-establishment transaction that was otherwise **acceptable** or **deprecated** at the time the transaction was initiated (also see Sec. 6 and 7)." If legacy use also extends to the use of derived keying material, then state so.
23	630	632	9.1.1	Conditions not specific enough. It shouldn't be acceptable for use in an unacceptable KA transaction.	Try: "The use of all other hash functions specified in [FIPS 180] and [FIPS 202] for one-step key derivation is **acceptable** (i.e., SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, and SHA3-512) when used in conjunction with a key-establishment transaction that was otherwise **acceptable** or **deprecated** at the time the transaction was initiated (see Section 6 and Section 7)."
23	634	634	9.1.1	Conditions not specific enough. It shouldn't be acceptable for use in an unacceptable KA transaction.	Try: "The use of KMAC128 and KMAC256 for one-step key derivation is **acceptable** when used in conjunction with a key-establishment transaction that was otherwise **acceptable** or **deprecated** at the time the transaction was initiated (see Section 6 and Section 7)."
23	638	639	9.1.2	Table 8 is erroneously called Table 9.	Replace "Table 9" by "Table 8" in this sentence.
23	640	640	9.1.2	Change the rightmost entry of the first row to reflect the <u>first-time use</u> status (i.e., during the initial execution of a KA transaction). Legacy use can be described in the text below the table.	New Table Entry: Deprecated through 2030; **Disallowed** after 2030
23	642	644	9.1.2	Conditions not specific enough.	Try: "The use of SHA-1 and the 224-bit hash functions (i.e., SHA-224, SHA-512/224,

					and SHA3-224) for two-step key derivation using HMAC during an otherwise **acceptable** or **deprecated** key-establishment transaction (see Sections 6 and 7) is **deprecated** through December 31, 2030, and **disallowed** for thereafter. (They are allowed for **legacy use** to re-derive keying material after 2030.) Further if legacy use extends to the use of derived keying material, then state so.
23	645	647	9.1.2	Conditions not specific enough.	Try: "The use all other hash functions specified in [FIPS 180] and [FIPS 202] (i.e., SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, and SHA3-512) for two-step key derivation using HMAC during an otherwise **acceptable** or **deprecated** key-establishment transaction (see Sections 6 and 7) is **acceptable.**
23	649	649	9.1.2	Replace number ("1.") by bullet.	
23	649	650	9.1.2	Conditions given are incorrect/incomplete and certainly not specific enough. Note: The extraction step can only produce a 128-bit KDK, so only AES-128 can be used in the key-expansion step.	Try: "During an otherwise **acceptable** or **deprecated** key-establishment transaction (see Sections 6 and 7), the use of AES-128, AES-192, or AES-256 as the cryptographic primitive for CMAC in the randomness-extraction step of two-step key derivation is **acceptable** only when AES-128 is used as the cryptographic primitive for CMAC in the key-expansion step."
24	656	657	9.2	In item 2, give the complete list of block-cipher flavors available.	Try: "2. CMAC, as specified in [SP 800-38B], requires the use of a block cipher algorithm (i.e., AES-128, AES-

					192, or AES-256, which are specified in [FIPS 197]).”
24	661	661	9.2	Table 9 is erroneously called Table 10.	Replace "Table 10" by "Table 9" in this sentence.
24	662	662	9.2	Change the rightmost entry of the first row to reflect the first-time use status.	New Table Entry: **Deprecated** through 2030; **Disallowed** after 2030 for derivation of new keying material. (**Legacy use** to re-derive keys is allowed after 2030.)
24	662	662	9.2	Change the rightmost entry of the 3rd row to reflect the first-time use status for CMAC w/ TDEA,	New Table Entry: **Disallowed** for deriving new keying material. (**Legacy use** to re-derive keys is allowed.)
24	664	666	9.2	Conditions not specific enough.	Try: “The use of SHA-1 and the 224-bit hash functions (i.e., SHA-224, SHA-512/224, and SHA3-224) for key derivation using HMAC is **deprecated** through December 31, 2030, and **disallowed** after 2030 for derivation of new keying material. (**Legacy use** to re-derive keys is allowed after 2030.)” Further if legacy use extends to the use of derived keying material, then state so.

24	671	674	9.2	Clarification possible.	Try: “The use of TDEA (as specified in [SP 800-67]) as the cryptographic primitive for CMAC is **disallowed** when initiating a (new) key-derivation transaction that uses a CMAC-based KDF. (**Legacy use** to re-derive previously derived keys is allowed.)” Further if legacy use extends to the use of derived keying material, then state so.
25	684	684	9.3	Table 10 is erroneously called Table 11.	Replace "Table 11" by "Table 10" in this sentence.
25	685	685	9.3	Change the rightmost entry of the first row to reflect the <u>first-time use</u> status.	Try: “**Deprecated** through 2030; **Disallowed** after 2030 for derivation of new keying material. (**Legacy use** to re-derive keys is allowed after 2030.)
24	687	689	9.3	Conditions not specific enough.	Try: “The use of SHA-1 and the 224-bit hash functions (i.e., SHA-224, SHA-512/224, and SHA3-224) for password-based key derivation using HMAC is **deprecated** through December 31, 2030, and **disallowed** after 2030 for derivation of new keying material. (**Legacy use** to re-derive keys is allowed after 2030.)” Further if legacy use extends to the use of derived keying material, then state so.
26	703	703	10	Table 11 is erroneously called Table 12.	Replace "Table 12" by "Table 11" in this sentence.
28	742	742	11	Table 12 is erroneously called Table 13.	Replace "Table 13" by "Table 12" in this sentence.
30	772	772	12	Table 13 is erroneously called Table 14.	Replace "Table 14" by "Table 13" in this sentence.
31	794	795	13	Table 14 is erroneously called Table 15.	Replace "Table 15" by "Table 14" in this sentence.

14. From: Aryeh Archer (SafeLogic), Date: December 4, 2024

#	Type	Line #s	Comment	Suggested Change
1.	E	308	For "The FF1 mode is acceptable…", acceptable should be in bold for consistency	"The FF1 mode is **acceptable**…",
2.	T	352 (Table 3), 359-385	Should these sections list an additional status for ECDSA using B and K curves? These curves are deprecated by FIPS 186-5. We believe their status would be: • *deprecated* for ECDSA generation (≥ 112 bits) • either *acceptable* or *legacy use* for ECDSA verification (≥ 112 bits)	1. Add rows to Table 3 to show separate statuses for ECDSA with P curves vs. ECDSA with B and K curves. 2. Add corresponding bullets to the status description in lines 359-385.
3.	E	352 (Table 3)	For "RSA generation (ASC X9.31)" and "RSA verification (ASC X9.31)", suggest changing "ASC X9.31" to "ANS X9.31" to align with FIPS 186 naming.	Update entries to: • "RSA generation (ANS X9.31)" • "RSA verification (ANS X9.31)"
4.	E	366-373, 378-385, 535-536, 541-545	Section 6 (lines 535-536 and 541-545) explicitly lists the brainpool curves from IG C.A. Consider also explicitly listing these curves in Section 3 (lines 360-373 and 374-385) for consistency.	List brainpool curves in Section 3 for consistency with Section 6.
5.	E	366-373, 378-385, 540	Section 6 (line 540) explicitly lists the secp256k1 curve from IG C.A. Consider also explicitly listing this curve in Section 3 (lines 360-373 and 374-385) for consistency.	List secp256k1 curve in Section 3 for consistency with Section 6.

15. From: Swapneela Unkule (ATSEC), Date: December 4, 2024

#	Line #	Comment (Include rationale for comment)	Suggested change. (if applicable)
1	Table 5	Key agreement not conformant to SP800-56A is now "legacy use". However, SP800-131Ar2 disallowed that after 2020, and the CMVP went to great lengths to have modules transition away from non-compliant SP800-56A key agreement schemes. Is the new legacy use really the intent of this document? It may have the effect of having those non-compliant schemes reappear in modules only to be again deprecated and disallowed within a few years.	
2	Table 6	Key agreement and key transport not compliant to SP800-56B was disallowed after 2020/2023 by SP800-133Ar2, and the CMVP enforced those transitions through IG. These schemes are now "legacy use" in this draft, which may have the effect of having those schemes reappear in modules only to be again deprecated and disallowed within a few years.	
3	577-578	"The processing of information in a key-establishment transaction is allowed for legacy use when len(n) = 1024 or len(q) = 160. See parameter set FA in [SP 800-56Ar2]." This seems to be a left-over, since this section talks about RSA and SP 800-56B. There is no relevant q value.	Remove the sentence.
4	Table 11	Combination of approved unauthenticated encryption and approved MAC for key wrapping becomes "deprecated", but unwrapping using the same combination remains "acceptable". Should unwrapping become "legacy use?"	Should unwrapping become "legacy use" instead of "acceptable"?
5	359	Should this list mention that elliptic curves over binary fields are considered deprecated by SP 800-186?	Add a new top-level bullet: "Signature generation and verification using elliptic curves over binary fields is **deprecated**."
6	424	Is there a minimum security strength that needs to be supported by keys that are generated? For example, would it be allowed to generate TDEA keys or RSA key pairs with a 1024-bit modulus?	Add minimum supported strength for generated key to be use as approved key.