Comments Received on SP 800-90A (December 2014)

From: Jim Nechvatal, NIST
Date: December 3, 2014

Nechvatal, 12/3/14

1. p. 1, Sec. 1, paragraph 3, line 4: "determined from the input from the randomness source" is vague. An input goes into something, not out of it. Also, it is a mystery as to why the important role of NRBGs in this context is virtually hidden (i.e., relegated to footnotes as on p. 26). It seems that this connection should be emphasized starting in Sec. 1. At a minimum, a reference to Sec. 8.6.5 should appear here.

2. p. 1, Sec. 1, paragraph 3, line 8: "sufficient entropy" is vague. A forward reference should be provided.

3. p. 1, Sec. 1, paragraph 3, last line: "instantiated security strength" is vague. A forward reference should be provided (e.g., to Sec. 8.4).

4. p. 2, Sec. 3, paragraph 3, line 1, end: "e.g." should probably be "i.e.".

5. p. 2, Sec. 3, paragraph 4: seems to contradict Sec. 1, line 1, which says "This Recommendation specifies techniques for the generation of random bits." How does "specifies techniques" differ from "structure, design and development"?

6. p. 4, definition of "DRBG Mechanism": "instantiation" is defined on p. 5, but "uninstantiate" is never explicitly defined. An entry should be added in Sec. 4, or a forward reference provided.

7. p. 5, definition of "Full Entropy": "epsilon" is superfluous here, and never explicitly used later. Thus, epsilon should be eliminated, and the definition re-stated as "at least (1-(2^-64))n bits". If epsilon is retained, it should be noted for what later purpose.

8. p. 5, definition of "Hash Function": "possibly very large" is meaningless unless defined rigorously.

9. p. 5, definition of "Health Testing": needs "." at end.

10. p. 5, definition of "Implementation Testing": line 2: "implemention" should be "implementation".

11. p. 6, definition of "Min-entropy": line 8: where did "n" come from?

12. p. 7, definition of "Random Number": lines 1-3 are vague because of the way the term "equal probability" is used. It should say that any two values have an equal probability of being chosen.

13. p. 7, definition of "Security Strength": line 2 refers to "operations of some sort"; line 7 refers to "basic operations". Both references are vague. At a minimum, they should be reconciled.

14. p. 9, definition of "leftmost(V,a)": should say "the leftmost a bits of bitstring V". If V is allowed to be a byte string, this needs to be clarified. Same for "rightmost" below.

14. p. 9, definition of "{a1,...,ai}": refers to "number of the ai", which makes no sense. Should define "{a1,...,an}".

15. p. 17, paragraph next to Fig. 3, line 16: "boundary or" should be "boundary, or"; non-parseable as is.

16. p. 20, Sec. 8.6.6, 1st paragraph, last line: "protected as well as the keys" could be parsed 2 ways: "inputs and seeds are protected to the same extent as the keys are protected", or "inputs and seeds are protected in addition to keys".

17. p. 20, Sec. 8.6.7, (a) and (b): "security_strength/2" in (a) is preferable to "1/2 security_strength" in (b). At a minimum, these should be reconciled, preferably in favor of (a). Similarly for "3/2 security_strength" at bottom of page.

18. p. 20, Sec. 8.6.7, component 4: line 2: "changes." should be "changes"; alternatively, "(For ... day.)" should be "For ... day.".

19. p. 24, Backtracking Resistance: last line: "with better than a 50-50 chance" only applies to ideal random sequences. It should be stated that an adversary has only a negligible advantage in prediction. Although it is not necessary, this could be formalized by saying that an adversary's probability of correctness is 1/2 + e, where e is a bounded security parameter, similar to the existing epsilon in the definition of full entropy on p. 5. Similar comment for Prediction Resistance below on p. 24.

20. p. 26, Sec. 9, paragraph 2, line 3: "envelope of pseudocode ... pseudocode in the envelopes" is vague. Does an envelope consist of pseudocode, or does it contain pseudocode? Also, "The pseudocode in the envelopes" implies that there are multiple envelopes all

containing the same pseudocode, which is presumably not the case.

21. p. 27, Sec. 9, paragraph beginning with "Consuming": line 3: "instantitate" should be "instantiate". Line 7: "heve" should be "have".

22. p. 31, bullet 1: line 2: "implemention" should be "implementation". line 5: "process step 6" should be "reseed process step 6" as in line 4.

23. p. 33, Sec. 9.3.1, bullet 1: line 2: "implemention" should be "implementation".

24. p. 33-34, Sec. 9.3.1., bullet 1: line 4: "process step 7.1" is vague. Same for "process step 10" on p. 34. In between, "generate process steps 1 and 7.3" is referenced, which is correct. All references should be to "generate process step(s) ...".

25. p. 36, step 11: formatting begins to deviate. On p. 30, "Return(SUCCESS,state_handle)" is used, which is correct. Here, "Return(SUCCESS and pseudorandom_bits)" is used; this is too loose. These should be reconciled in favor of p. 30.

26. p. 36, Implementation notes, line 2: "removed; and" should be "removed, and".

27. p. 38, Uninstantiate_function, line 2: "implemention" should be "implementation".

28. p. 38, Uninstantiate_function, line 4: "process": what process?

29. p. 39, Sec. 10.1, line 1: "hash function that is non-invertible or one-way" is ambiguous; "or" could mean "either-or". "non-invertible" is not defined on p. 5; only "one-way" is defined.

30. p. 39, Sec. 10.1, paragraph 3, line 3: "57]" should be "57".

31. p. 40, Table 2 continued: the 1st entry is both ambiguous and erroneous. Note that on p. 39, the last table states that "min_length = security_strength", which is unambiguous (and apparently correct). In the first entry on p. 40, "max_length" is specified as "<= 2^35 bits". Interpreted literally this means "The maximum of the maximum of the entropy input length is 2^35 bits", which is gibberish. If the intent is to say "max_length = 2^35 bits", then "<= 2^35" should be "2^35". On the other hand, if max_length is intended to be a variable, the entry is incorrect: in

light of the previous entry on p. 39, the 1st entry on p. 40 should be "security_stength <= maxlength <= 2^35 bits". The latter would imply that "max_length" is a user-defined or implementation-dependent parameter value. If this is the case, text should be added explaining the difference between the constant min_length and the variable max_length, and stating how, and by whom, max_length is set. A similar discussion applies to other entries where "<=" appears.

32. p. 40, Sec. 10.1.1.1, bullets (a)- (c): "constant C" in (b) should be "constant (C)" corresponding to "value (V)" in (a) (cf. (c)).

33. p. 41, Sec. 10.1.1.2, "Notes for the instantiate function", line 4: "Process step 9 of that function" is vague. The instantiate function defined on pp. 27-28 has 3 steps. Note that on p. 41, "The instantiate algorithm", paragraph 2, line 2 refers to "step 9 of the instantiate process in Section 9.1", which is correct. The latter reference should be used in both instances. Similar changes should be made in Sec. 10.1.1.3, 10.1.1.4, 10.1.2.3 - 10.1.2.5., and 10.2.1.3 - 10.2.1.5.

34. p. 42, "Hash_DRBG Instantiate Process", step 6: further deterioration of notation. On pp. 30,33,38, the correct notation "Return(parameter,...,parameter)" is used. From here to p. 71, the notation changes to the less-precise "Return <informal list of parameters>". Step 6 on p. 42 should be "initial_working_state = (V,C, reseed_counter)". Same comment applies on pp. 43,44,47-49, 54-59,61-64, and 71. Note that on p. 77, "Return("Success", state_handle) appears, indicating a return to correct notation, which is used through the remainder of the paper.

35. p. 44, "Hash_DRBG Generate Process": step 3: "Hashgen" does not appear to be defined, in contrast to "Hash" and "Hash_DRBG", which are defined.

36. p. 44, "Hashgen Process", step 4: defines the sequence {wi}, which greatly increases space allocation (array needed). {wi} is never used, so it should be replaced by a simple variable "w".

37. p. 46, 1st full paragraph: line 1: "value of V and Key" should be "values of V and Key".

38. p. 47, Sec 10.1.2.3, title: needs blank space after "10.1.2.3".

39. p. 49, step 8 at bottom: if change indicated in comment 34 above is not made, then "_state)" should be "_state".

40. p. 51, Table 3: same comments apply as in comment 31 above for entries with "<=".

41. p. 51, Table 3, "If a derivation function is not used", 2nd entry: "_length)(blocklen" should be "_length) = (blocklen".

42. p. 52, top, 1st table entry: "- 4^)" should be "- 4)".

43. p. 52, paragraph 2, line 14: "provide" should be "provides".

44. p. 54, step 2.1: the "else" branch is ambiguous. Why would the single instruction be labeled "2.1.2"? This might suggest (incorrectly) that instruction 2.1.1 has been inadvertently omitted from the "else" branch. It would be better to use the format of p. 60, "CTR_DRBG Generate Process", step 2, where the "else" branch is unambiguous. A similar comment applies on p. 59, step 4.1.

45. p. 54, Sec. 10.2.1.3.1: title should probably be "Instantiation when ...".

46. p. 55, 2nd step 3: the exclusive-or only makes sense if entropy_input has length exactly seedlen. This does not seem to be made explicit. If it is implicit, the entropy_input returned by the randomness source should be checked for correct length. A check should be inserted into the pseudocode before step 3. A similar comment applies on p. 57, 2nd step 3.

47. p. 58, Sec. 10.2.1.5.1, title: why is "Not" underlined (apparently for emphasis), as opposed to 10.2.1.4.1? Similarly for "Is" in the title of 10.1.2.5.2.

48. p. 60, Sec. 10.2.1.5.2, paragraph 2: "Let df be a derivation function" is ambiguous from an implementer's point of view. If the intent is for the implementer to use each of the two df's in Sec. 10.3, thereby producing two schemes for generating pseudorandom bits, the question is why: most implementations will only require one scheme. So most likely the intent is for an implementer to select one of the two df's. But how? By coin flip? The issue of number of df's should be clarified; assuming the intended number is one, guidance should be provided for the selection process.

49. p. 63, steps 2 and 3, line 1: "represention" should be "representation".

50. p. 63, step 5: "outlen) != 0, S =" should be "outlen) != 0 do
S =" as in steps 9 and 13.

51. p. 63, step 11: "Next outlen bits" is ambiguous.

52. p. 64, "Block_Encrypt", paragraph beginning with "For TDEA":
line 3: "197]" should be "197".

53. p. 65, Sec. 11, last paragraph, line 2: "knowingly" is too vague a qualifier to follow "shall not"; it opens too wide a loophole. This seems to be the unique occurrence of "knowingly", so why here?

54. p. 66, 2nd bullet, line 1: "implemention" should be "implementation".

55. p. 67, Sec. 11.3.1, last paragraph, line 2: "first-use" should be "first use".

56. p. 68, Sec. 11.3.6.1, 1st paragraph, lines 5-7: unparseable. In line 5, "invalid," should be "invalid" to remove the ambiguity.

57. p. 70, App. A.1: inputs should be clearly defined, which they are not. "Process" refers to "b", which is not defined.

58. P. 70, Apps A.1,A.2: in A.1, bits are labeled "from leftmost to rightmost". In A.2 they are labeled "most significant to least significant". These should be reconciled.

59. p. 70, App. A.2: inputs should be clearly defined, which they are not. Output is b1,..., bn, which is undefined: step 2 of "Process" refers to "any integer n", so n is undefined. There may be a connection with the sentence appearing just before App. A.2; if so, this needs to be explicit, and the "any integer n" deleted.

60. p. 70, bottom sentence: duplicates the sentence just before App. A.2. Also, where is this used?

61. p. 71, App. A.3, "Integer_to_byte_string(x)": what does "intended length n ... satisfying 2^8n > x" mean? This is ambiguous. Since "n" is not well-defined, neither is the output "O" (remark: "O" is easily confused with "0"; "B" would be better to label bytes).

62. p. 71, Apps. A.3,A.4, "Process" step 2: sums have no limits (the "for i = 1 to n" below is extraneous), in contrast to step 2 in Apps. A.1,A.2. Limits should be added.

63. p. 71, App. A.3, "Output": what does "O" have to do with the input "x"? In App. A.4, "Output", what does "x" have to do with "O"? These questions were answered in the "Output" in Apps. A.1.,A.2.

64. p. 71, App. A.5, 1st sentence: incomprehensible. It should read "In some cryptographic applications, a sequence of n random numbers (a0,...,a(n-1)) is required, where n is a positive integer and:" Using this notation, in (ii) below, "i >= 0" should be "for any i (0 <= i <= n-1)". Introduction of "n" is necessary because no cryptographic application requests an infinite sequence of random numbers (an indefinitely long stream could be needed, but the techniques in A.5.1 - A.5.4 produce sequences of definite length).

65. p. 72, App. A.5.1: 1st sentence: what is "r"? It cannot be inferred from App. A.5, since "r" there could be arbitrary or a derived value (r=b-a+1). Also, what is the relation between m,r and a? This should be explicit. Also, App. A.5 dealt with a sequence of random numbers. Why does A.5.1 shift to the creation of a single random number?

66. p. 72, Apps A.5.1,A.5.2: use and relation of these are ambiguous. A.5.2 is billed as an "alternative" to A.5.1. However, A.5.1 returns a single random number; A.5.2 returns t numbers, so they are generally incomparable. They are only comparable when t = 1; as noted on p. 73, the routines coincide in this event. The relationship of the routines, and the relations among m,r,a and t and the random numbers computed needs to be explained much more thoroughly for these sections to be useful.

67. p. 73, Apps A.5.3,A.5.4: see comments 65,66 above.

68. p. 77, step 8: "Hash_df" is not explicitly defined in App. B. If it is implicitly defined, this should be indicated.

69. p. 79, step 6: "OR" should be "or" as in step 1. Same for p. 89, step 7.

70. p. 80, comment after step 6.4: "Hashgen" is not explicitly defined in App. B. If it is implicitly defined, this should be indicated.

71. p. 80, step 8: "bitd" should be "bits".

72. p. 80, step 11: see comment 36 above.

73. p. 82, step 3: each of the two "Else (requested" should be
"Else if (requested". Meaningless as currently stated.

74. p. 91, bottom line: the exclusive-or only makes sense if
"entropy_input" has length exactly equal to the length of
"personalization_string". This does not seem to be
made explicit, in App. B.3.2 or B.4.2. If it is implicit,
a check should be inserted into the pseudocode before the
bottom line. A similar comment applies on p. 92, App. B.4.3,
last line.

75. p. 93, bullet (c), paragraph 2, line 2: "parallize" should be
"parallelize".

76. p. 94, "Security": lines 3 and 5 refer to "random oracle" and
"pre-image resistance", which have not been defined. Entries
should be added in Sec. 4.

77. p. 98, 1st sentence: "this original" should be "The original".
Also, "June, 2006" should be "June 2007" (or change "March 2007").

78. p. 98, bullet 3, line 2: "privater" should be "private".

79. p. 99, bullet 6, line 2: "reeeding" should be "reseeding".

80. p. 99, bullet 10, line 2: "requst" should be "request".

81. p. 99, bullet 11, 2nd unlabeled subbullet: line 2: "during
instantiation." should be "during instantiation:". Otherwise the
2nd paragraph is dangling.

82. p. 100, bullet 12, 5th unlabeled subbullet: needs a re-write.

83. p. 100, bullet 13, 1st unlabeled subbullet: what is "item"?

84. p. 100, bullet 16, bullet 16, line 3: "ro" should be "as".

85. p. 102, bullet 1, line 2: "resistence" should be "resistance".

86. p. 102, bullet 2: incomprehensible. For starters, in line 2,
"with between" needs change. Beyond this, who knows.

87. p. 102, bullet 4, line 3: "curves, the old" should be "curves and
the old".

88. p. 103, bullet 6, line 7: "teh nsecond" should be "the second".

89. p. 103, bullet 11, line 1: "9:A" should be "9: A".

90. p. 103, bullet 12, line 2: a guess: maybe "for explanatory" should be "has been added for explanatory".

91. p. 104, bullet 12, line 7: "(note that ...": on and on and on it goes; where it stops nobody knows.

92. p. 104, bullet 15, line 3: "disuccion" should be "discussion".

93. p. 104, last line: "vaidated" should be "validated".

From: Bluma Sussman, Contractor Support to CE&A for DHS
Date: December 12, 2014

Attached are the DHS Cybersecurity Education & Awareness Branch (CE&A) recommendations for NIST SP 800-90 A, Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators. Our main focus, Attachment 1, was to provide a draft appendix for roles and responsibilities and reference the National Cybersecurity Workforce Framework (Workforce Framework). Some additional references are recommended (in yellow) for Appendix D (Attachment 2).

(Attachment 1):
APPENDIX XYZ
ROLES AND RESPONSIBILITIES
RECOMMENDATION FOR RANDOM NUMBER GENERATION USING DETERMINISTIC RANDOM BIT GENERATORS
Referencing the National Cybersecurity Workforce Framework ("Workforce Framework"), this appendix provides examples of the roles and responsibilities of possible stakeholders associated with various cryptography techniques. Additional cybersecurity roles and responsibilities are included within the Workforce Framework, located on the National Initiative for Cybersecurity Careers and Studies (NICCS™) Portal (http://niccs.us-cert.gov) and the NIST website (http://csrc.nist.gov/nice/framework). The Workforce Framework is the foundation for increasing the size and capability of the US Cybersecurity Workforce.

Deterministic Random Bit Generator (DRBG) implementation conformance testing involves multiple roles throughout an organization. Each organization may involve several subordinate and partner organizations to plan and design DRBG solutions with stakeholder roles and responsibilities varying. The following roles are listed as a guide. Depending on the organization's needs, a stakeholder may be assigned one of the roles listed below or a combination of roles relevant to DRBG techniques. In smaller organizations, a single individual may hold multiple roles.

Chief Information Officer (CIO). A senior-level executive who directs, plans, organizes, and controls all activities of the Management Information Systems (MIS) Department to ensure the effective, efficient, and secure operation of automated data processing systems.

Chief Security Officer (CSO) or Chief Information Security Officer (CISO). A senior-level executive responsible for establishing and maintaining the enterprise vision, strategy, and program to ensure information assets and technologies are adequately protected. The CSO/CISO directs staff in identifying, developing, implementing, and maintaining organization processes to reduce information and communications technology (ICT) risks. The CSO/CISO responds to incidents, establishes appropriate standards and controls, manages security technologies, and directs the establishment and implementation of policies and procedures.

Chief Technology Officer (CTO). A senior-level executive who oversees all technical characteristics of an organization. The CTO works directly with senior-level executives to grow the organization through the use of technological resources. The CTO directs all staff in information technology (IT) and information operations (IO) departments to achieve the organization's strategic goals that are established in a strategic plan.
Configuration Manager. Establishes and maintains the integrity of information and communications technology (ICT) products and information systems through control of processes for initializing, changing, and monitoring the configurations of those products and systems throughout the system development life cycle.

Information Assurance Architect and Engineer. Designs, develops, implements, and/or integrates an organization's information assurance architecture, system, or system component for use within their computing, network, and enclave environments. Ensures the architecture and design of an information system are functional and secure. May also be responsible for system or network designs that encompass multiple computing environments and/or network environments to include those with differing data protection/classification requirements.

Network Security Analyst. Manages security efforts of computer networks and information systems. This includes auditing the network for vulnerabilities, developing solutions for security concerns, and investigating security breaches. Often responsible for the training and education of an organization's staff in computer security best practices.
Research & Development Engineer. Conducts technology and/or feasibility assessments while facilitating innovation. Provides, builds, and supports a prototype capability and/or evaluates its security and utility.

Security Architect. Designs a security system or major components of a security system, and may lead a security design team building a new security system.
Systems Engineer. Designs and develops software, computer systems, and networks. Preserves safety and security of the computing environments. Ensures network and computer systems in an organization are working properly and secured from hacking and virus attacks.

Systems Security Analyst. Analyzes and assesses vulnerabilities in the infrastructure (e.g., software, hardware, networks), investigates available tools and countermeasures to remedy the detected vulnerabilities, and recommends solutions and best practices. Analyzes and assesses damage to the data/infrastructure as a result of security incidents, examines available recovery tools and processes, and recommends solutions. Conducts tests for compliance with security policies and procedures. May assist in the creation, implementation, and/or management of security solutions.

(Attachment 2)
APPENDIX D
REFERENCES

National Initiative for Cybersecurity Careers and Studies (NICCS™) Website, *www.niccs.us-cert.gov*.

The National Initiative for Cybersecurity Education (NICE), *http://csrc.nist.gov/nice*.

National Cybersecurity Workforce Framework, *www.niccs.us-cert.gov/training/tc/framework*, and *http://csrc.nist.gov/nice/framework*, released in 2011.

From: Robert Burns, Thales Security
Date: December 17, 2014

[Legend (type of comment); E = Editorial; G = General; T = Technical]

ID	Sect., Subj., & Para.	Type	Comment	Recommendation
001	General	G	Figures (especially) 1 – 3 are not clear and appear rasterized. Figure 11 has illegible text against the V input.	Recommend the use of high-resolution images or utilization of vector format for the diagrams.
002	Section 8.5	E	Although the figures clearly illustrate the logical notion of a DRBG Mechanism Boundary, the illustrations introduce some ambiguity with respect to the relationship to the cryptographic boundary. Although this is largely delegated to the SP800-90C standard, the use of the term cryptographic boundary in this context should be made unambiguous.	Recommend adding cryptographic boundary representations to the figures to disambiguate the relationship between the cryptographic module boundaries, DRBG mechanism boundaries, and sub-boundaries.
003	Section 8.6	E	Section 8.6 asserts that entropy input (e.g. nonce) must be obtained from within a cryptographic module boundary. This could be problematic if the module was a software implementation/library that got entropy from an external source. This assertion precludes the use of other sources of trusted entropy which may have come from outside the cryptographic boundary.	Recommend clarifying statements around external nonce entropy sources. Specifically concerning the use of another hardware source outside the module and/or pre-generated entropy. We believe the use of entropy from trusted external sources should be permitted.
004	Section 10.2.1	T	Research by Matthew J. Campagna (http://eprint.iacr.org/2006/379.pdf) has indicated that the NIST CTR DRBG has lower security	Recommendation is to update the section on CTR DRBGs reflecting the more conservative security estimates provided in the

			strengths than those indicated by the strengths of the underlying cipher suite. Although NIST have attempted to address the claims made in the Campagna paper with the comment from Section 10.2.1 in regards to limiting the total number of generation requests and bits per request, we believe that this falls short for the following reasons: 1. It creates an interdependence between the three standards which can be avoided. 2. It violates the principle that a composition of cryptographic primitives is only as strong as its weakest component. 3. It complicates the security analysis, (e.g. How are we justifying the reseed limits with respect to each cipher?).	Campagna paper OR provide security analysis justifying the imposed limits so that the impact of these limits can be validated independently.
005	Section 8.6.7	T	In reference to the use of a timestamp, the assertion is made that, "For case 2 above, the timestamp must be trusted. A trusted timestamp is generated and signed by an entity that is trusted to provide accurate time information." Although we agree the timestamp must be trusted, the assertion that it must be signed by another entity is only one solution to the problem of trusted time. For example, a secure module may have an internal time source which is trusted and would not require signature.	Recommend removing the sentence, "A trusted timestamp is generated and signed by an entity that is trusted to provide accurate time information."

From: Deepnarayan Choubey <deep_choubey2009@yahoo.com>
Date: December 22, 2014

Good article for random work. Thanks to all.

From: NSA
Date: December 23, 2014

Section 4. The definition of randomness source is incomplete. As explained in the change log (#2 on p. 102) “randomness source” could be an entropy source, a NRBG, or a DRBG. As it is currently written, the definition of randomness source only states DRBG. The definition needs to include NRBGs and entropy sources. It is consistent with the rest of the document to include this expanded definition of randomness source.

Section 7. Why is Figure 1 so small? Many of the figures in the document should be a bit larger so they are easier to read, but this one seems especially small.

From: Sonu Shankar, Cisco
Date: December 23, 2014

1) NIST has replaced references to “entropy input” with “randomness source” in this draft. There are also references now to an “approved randomness source”, e.g. Chapter 7 “Functional Model of a DRBG”. Seeing as SP 800-90B is still currently in draft form, we would like some clarity on what it would take to deem a randomness source as “approved”. Adding clear guidance to 800-90A on this topic is especially critical considering the definition has now expanded to include an entropy source (in the traditional sense), an NRBG or a DRBG.

2) As the “randomness source” has been defined as a component of the DRBG, we would like some clarity on the impact of positioning the randomness source in the context of the DRBG mechanism boundary and cryptographic module boundary definitions. We suggest adding clear guidance to 800-90A on approved architectures for the mechanism boundary, considering the practical case of software-based DRBG implementations part of software cryptographic libraries depending on hardware-based entropy (randomness sources). What needs to be contained within the mechanism boundary and what can exist outside in the cryptographic module boundary is critical here.

3) From section 8.6.7, “Each nonce shall be unique to the cryptographic module in which instantiation is performed, but need not be secret. When used, the nonce shall be considered to be a critical security parameter.” Does NIST really mean to suggest that the nonce shall be considered a critical security parameter? The first line suggests that it need not be secret. Please clarify.

From: Michael Harris, CDC/OCOO/OCIO
Date: December 30, 2014

CDC has no comments to provide on the *DRAFT SP 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators*.

From: ^OIS Controls, SSA
Date: December 30, 2014

The Social Security Administration (SSA) appreciates the opportunity to review and provide comments on the National Institute of Standards and Technology DRAFT SP 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators.

SSA has no comment on this draft.