

# Withdrawn Draft

## Warning Notice

The attached draft document has been withdrawn, and is provided solely for historical purposes. It has been superseded by the document identified below.

**Withdrawal Date** July 23, 2019

**Original Release Date** March 6, 2019

## Superseding Document

**Status** Final

**Series/Number** NIST Special Publication 800-133 Revision 1

**Title** Recommendation for Cryptographic Key Generation

**Publication Date** July 2019

**DOI** <https://doi.org/10.6028/NIST.SP.800-133r1>

**CSRC URL** <https://csrc.nist.gov/publications/detail/sp/800-133/rev-1/final>

**Additional Information** Key Establishment

<https://csrc.nist.gov/projects/key-management/key-establishment>

---

3  
4  
5 **Recommendation for Cryptographic**  
6 **Key Generation**

---

7  
8  
9 Elaine Barker  
10 Allen Roginsky

11  
12  
13  
14  
15  
16  
17  
18 This publication is available free of charge from:  
19 <https://doi.org/10.6028/NIST.SP.800-133r1-draft>

---

20  
21  
22  
23 C O M P U T E R S E C U R I T Y

---

25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51

**Draft NIST Special Publication 800-133**  
**Revision 1**

**Recommendation for Cryptographic**  
**Key Generation**

Elaine Barker  
Allen Roginsky  
*Computer Security Division*  
*Information Technology Laboratory*

This publication is available free of charge from:  
<https://doi.org/10.6028/NIST.SP.800-133r1-draft>

March 2019



U.S. Department of Commerce  
*Wilbur L. Ross, Secretary*

National Institute of Standards and Technology  
*Walter Copan, NIST Director and Under Secretary of Commerce for Standards and Technology*

52  
53  
54  
55  
56  
57  
58  
59

60

## Authority

61 This publication has been developed by NIST in accordance with its statutory  
62 responsibilities under the Federal Information Security Modernization Act (FISMA) of  
63 2014, 44 U.S.C. § 3551 *et seq.*, Public Law (P.L.) 113-283. NIST is responsible for  
64 developing information security standards and guidelines, including minimum  
65 requirements for federal information systems, but such standards and guidelines shall not  
66 apply to national security systems without the express approval of appropriate federal  
67 officials exercising policy authority over such systems. This guideline is consistent with  
68 the requirements of the Office of Management and Budget (OMB) Circular A-130.

69 Nothing in this publication should be taken to contradict the standards and guidelines made  
70 mandatory and binding on federal agencies by the Secretary of Commerce under statutory  
71 authority. Nor should these guidelines be interpreted as altering or superseding the existing  
72 authorities of the Secretary of Commerce, Director of the OMB, or any other federal  
73 official. This publication may be used by nongovernmental organizations on a voluntary  
74 basis and is not subject to copyright in the United States. Attribution would, however, be  
75 appreciated by NIST.

76 National Institute of Standards and Technology Special Publication 800-133 Revision 1  
77 Natl. Inst. Stand. Technol. Spec. Publ. 800-133 Rev 1, 27 pages (March 2019)  
78 CODEN: NSPUE2

79 This publication is available free of charge from:  
80 <https://doi.org/10.6028/NIST.SP.800-133r1-draft>

81 Certain commercial entities, equipment, or materials may be identified in this document in order to describe  
82 an experimental procedure or concept adequately. Such identification is not intended to imply  
83 recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment  
84 are necessarily the best available for the purpose.

85 There may be references in this publication to other publications currently under development by NIST in  
86 accordance with its assigned statutory responsibilities. The information in this publication, including  
87 concepts and methodologies, may be used by federal agencies even before the completion of such companion  
88 publications. Thus, until each publication is completed, current requirements, guidelines, and procedures,  
89 where they exist, remain operative. For planning and transition purposes, federal agencies may wish to  
90 closely follow the development of these new publications by NIST.

91 Organizations are encouraged to review all draft publications during public comment periods and provide  
92 feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at  
93 <https://csrc.nist.gov/publications>.

94 **Public comment period: March 6, 2019 through May 8, 2019**

95  
96 National Institute of Standards and Technology  
97 Attn: Computer Security Division, Information Technology Laboratory  
98 100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930  
99 Email: [SP-800-133\\_Comments@nist.gov](mailto:SP-800-133_Comments@nist.gov)

100

101 All comments are subject to release under the Freedom of Information Act (FOIA)

102

## Reports on Computer Systems Technology

103 The Information Technology Laboratory (ITL) at the National Institute of Standards and  
104 Technology (NIST) promotes the U.S. economy and public welfare by providing technical  
105 leadership for the Nation's measurement and standards infrastructure. ITL develops tests,  
106 test methods, reference data, proof of concept implementations, and technical analyses to  
107 advance the development and productive use of information technology. ITL's  
108 responsibilities include the development of management, administrative, technical, and  
109 physical standards and guidelines for the cost-effective security and privacy of other than  
110 national security-related information in Federal information systems. The Special  
111 Publication 800-series reports on ITL's research, guidelines, and outreach efforts in  
112 information system security, and its collaborative activities with industry, government, and  
113 academic organizations.

114

115

### Abstract

116 Cryptography is often used in an information technology security environment to protect  
117 data that is sensitive, has a high value, or is vulnerable to unauthorized disclosure or  
118 undetected modification during transmission or while in storage. Cryptography relies upon  
119 two basic components: an algorithm (or cryptographic methodology) and a cryptographic  
120 key. This Recommendation discusses the generation of the keys to be managed and used  
121 by the **approved** cryptographic algorithms.

122

123

### Keywords

124 asymmetric key; key agreement; key derivation; key generation; key wrapping; key  
125 replacement; key transport; private key; public key; symmetric key.

126

127

### Acknowledgements

128 The National Institute of Standards and Technology (NIST) gratefully acknowledges and  
129 appreciates contributions by Rich Davis of the National Security Agency and the public  
130 and private sectors whose thoughtful and constructive comments improved the quality and  
131 usefulness of this publication.

132

133

### Note to Reviewers

134 Sections [4](#) and [5.1](#) mention the use of EdDSA for the generation of digital signatures.  
135 EdDSA will be included in a proposed revision of FIPS 186.

136

137

**Call for Patent Claims**

138 This public review includes a call for information on essential patent claims (claims  
139 whose use would be required for compliance with the guidance or requirements in this  
140 Information Technology Laboratory (ITL) draft publication). Such guidance and/or  
141 requirements may be directly stated in this ITL Publication or by reference to another  
142 publication. This call also includes disclosure, where known, of the existence of pending  
143 U.S. or foreign patent applications relating to this ITL draft publication and of any  
144 relevant unexpired U.S. or foreign patents.

145 ITL may require from the patent holder, or a party authorized to make assurances on its  
146 behalf, in written or electronic form, either:

147 a) assurance in the form of a general disclaimer to the effect that such party does not  
148 hold and does not currently intend holding any essential patent claim(s); or

149 b) assurance that a license to such essential patent claim(s) will be made available to  
150 applicants desiring to utilize the license for the purpose of complying with the  
151 guidance or requirements in this ITL draft publication either:

152 i. under reasonable terms and conditions that are demonstrably free of any  
153 unfair discrimination; or

154 ii. without compensation and under reasonable terms and conditions that are  
155 demonstrably free of any unfair discrimination.

156 Such assurance shall indicate that the patent holder (or third party authorized to make  
157 assurances on its behalf) will include in any documents transferring ownership of patents  
158 subject to the assurance, provisions sufficient to ensure that the commitments in the  
159 assurance are binding on the transferee, and that the transferee will similarly include  
160 appropriate provisions in the event of future transfers with the goal of binding each  
161 successor-in-interest.

162 The assurance shall also indicate that it is intended to be binding on successors-in-interest  
163 regardless of whether such provisions are included in the relevant transfer documents.

164 Such statements should be addressed to: [SP-800-133\\_Comments@nist.gov](mailto:SP-800-133_Comments@nist.gov)

165

166

167

**Table of Contents**

168	1	Introduction.....	1
169	2	Definitions, Acronyms and Symbols.....	1
170	2.1	Definitions.....	1
171	2.2	Acronyms.....	6
172	2.3	Symbols.....	7
173	3	General Discussion .....	7
174	3.1	Keys to Be Generated .....	7
175	3.2	Where Keys are Generated .....	8
176	3.3	Supporting a Security Strength .....	8
177	4	Using the Output of a Random Bit Generator.....	9
178	5	Generation of Key Pairs for Asymmetric-Key Algorithms.....	11
179	5.1	Key Pairs for Digital Signature Schemes.....	11
180	5.2	Key Pairs for Key Establishment.....	11
181	5.3	Distributing the Key Pairs.....	12
182	5.4	Key Pair Replacement.....	12
183	6	Generation of Keys for Symmetric-Key Algorithms .....	13
184	6.1	The “Direct Generation” of Symmetric Keys.....	14
185	6.2	Distributing the Generated Symmetric Key.....	14
186	6.3	Symmetric Keys Generated Using Key-Agreement Schemes.....	14
187	6.4	Symmetric Keys Derived from a Pre-shared Key.....	15
188	6.5	Symmetric Keys Derived from Passwords .....	15
189	6.6	Symmetric Keys Produced by Combining Multiple Keys and Other Data .....	16
190	6.7	Replacement of Symmetric Keys.....	16
191		Appendix A: References.....	18
192		Appendix B: Revisions .....	21
193			

194 **1 Introduction**

195 Cryptography is often used in an information technology security environment to protect  
196 data that is sensitive, has a high value, or is vulnerable to unauthorized disclosure or  
197 undetected modification during transmission or while in storage. Cryptography relies upon  
198 two basic components: an algorithm (or cryptographic methodology) and a cryptographic  
199 key. The algorithm is a mathematical function, and the key is a parameter used by that  
200 function.

201 The National Institute of Standards and Technology (NIST) has developed a wide variety  
202 of Federal Information Processing Standards (FIPS) and NIST Special Publications (SPs)  
203 to specify and approve cryptographic algorithms for Federal government use. In addition,  
204 guidance has been provided on the management of the cryptographic keys to be used with  
205 these **approved** cryptographic algorithms.

206 This Recommendation discusses the generation of the keys to be used with the **approved**  
207 cryptographic algorithms. The keys are either generated using mathematical processing on  
208 the output of **approved** Random Bit Generators and possibly other parameters or generated  
209 based upon keys that are generated in this fashion.

210 **2 Definitions, Acronyms and Symbols**211 **2.1 Definitions**

<b>Approved</b>	FIPS-approved and/or NIST-recommended.
Asymmetric key	A cryptographic key used with an asymmetric-key (public-key) algorithm. The key may be a private key or a public key.
Asymmetric-key algorithm	A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that determining the private key from the public key is computationally infeasible. Also known as a public-key algorithm.
Bit string	An ordered sequence of 0 and 1 bits.
Ciphertext	Data in its encrypted form.
Compromise	The unauthorized disclosure, modification or use of sensitive data (e.g., keying material and other security-related information).
Cryptographic algorithm	A well-defined computational procedure that takes variable inputs, often including a cryptographic key, and produces an output.

Cryptographic key (key)	<p>A parameter used in conjunction with a cryptographic algorithm that determines its operation in such a way that an entity with knowledge of the key can reproduce or reverse the operation, while an entity without knowledge of the key cannot. Examples of cryptographic operations requiring the use of cryptographic keys include:</p> <ol style="list-style-type: none"> <li>1. The transformation of plaintext data into ciphertext data,</li> <li>2. The transformation of ciphertext data into plaintext data,</li> <li>3. The computation of a digital signature from data,</li> <li>4. The verification of a digital signature,</li> <li>5. The computation of an authentication code from data,</li> <li>6. The verification of an authentication code from data and a received authentication code,</li> <li>7. The computation of a shared secret that is used to derive keying material.</li> <li>8. The derivation of additional keying material from a key-derivation key (i.e., a pre-shared key).</li> </ol>
Cryptographic module	The set of hardware, software, and/or firmware that implements security functions (including cryptographic algorithms and key generation) and is contained within a cryptographic module boundary. See <a href="#">FIPS 140</a> . <sup>1</sup>
Cryptographic boundary	An explicitly defined continuous perimeter that establishes the physical bounds of a cryptographic module and contains all the hardware, software and/or firmware components of a cryptographic module. See <a href="#">FIPS 140</a> .
Cryptoperiod	The time span during which a specific key is authorized for use or in which the keys for a given system or application may remain in effect.
Data integrity	A property possessed by data items that have not been altered in an unauthorized manner since they were created, transmitted or stored.
Decryption	The process of changing ciphertext into plaintext using a cryptographic algorithm and key.

---

<sup>1</sup> FIPS 140: *Security Requirements for Cryptographic Modules*.

Digital signature	The result of a cryptographic transformation of data that, when properly implemented, provides origin authentication, assurance of data integrity and supports signatory non-repudiation.
Encryption	The process of changing plaintext into ciphertext using a cryptographic algorithm and key.
Entity	An individual (person), organization, device or process. Used interchangeably with “party”.
Entropy	A measure of the disorder, randomness or variability in a closed system. See <a href="#">SP 800-90B</a> . <sup>2</sup>
Key	See cryptographic key.
Key agreement	A (pair-wise) key-establishment procedure in which the resultant secret keying material is a function of information contributed by both participants, so that neither party can predetermine the value of the secret keying material independently from the contributions of the other party. Contrast with key transport.
Key-agreement primitive	A primitive algorithm used in a key-agreement scheme specified in <a href="#">SP 800-56A</a> , <sup>3</sup> or in <a href="#">SP 800-56B</a> . <sup>4</sup>
Key derivation	<ol style="list-style-type: none"> <li>1. A process by which one or more keys are derived from a shared secret and other information during a key agreement transaction.</li> <li>2. A process that derives new keying material from a key (i.e., a key-derivation key) that is currently available.</li> </ol>
Key-derivation key	A key used as an input to a key-derivation method to derive other keys. See <a href="#">SP 800-108</a> . <sup>5</sup>
Key establishment	A procedure that results in secret keying material that is shared among different parties.
Key-generating module	A cryptographic module in which a given key is generated.
Key generation	The process of generating keys for cryptography.
Key pair	A private key and its corresponding public key; a key pair is used with an asymmetric-key (public-key) algorithm.

<sup>2</sup> SP 800-90B, *Recommendation for the Validation of Entropy Sources for Random Bit Generation*.

<sup>3</sup> SP 800-56A: *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*.

<sup>4</sup> SP 800-56B: *Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography*.

<sup>5</sup> SP 800-108: *Recommendation for Key Derivation Using Pseudorandom Functions*.

Key-pair owner	For an asymmetric-key algorithm, the entity that is authorized to use the private key associated with a public key, whether that entity generated the key pair itself or a trusted party generated the key pair for the entity.
Key transport	A key-establishment procedure whereby one party (the sender) selects a value for the secret keying material and then securely distributes that value to another party (the receiver) using an asymmetric algorithm.
Key wrapping	A method of encrypting and decrypting keys and (possibly) associated data using a symmetric key; both confidentiality and integrity protection are provided. See <a href="#">SP 800-38F</a> . <sup>6</sup>
Module	See Cryptographic module.
Origin authentication	A process that provides assurance of the origin of information (e.g., by providing assurance of the originator's identity).
Owner	<ol style="list-style-type: none"> <li>1. For an asymmetric key pair, consisting of a private key and a public key, the owner is the entity that is authorized to use the private key associated with a public key, whether that entity generated the key pair itself or a trusted party generated the key pair for the entity.</li> <li>2. For a symmetric key (i.e., a secret key), the entity or entities that are authorized to share and use the key.</li> </ol>
Party	See Entity.
Password	A string of characters (letters, numbers and other symbols) that are used to authenticate an identity or to verify access authorization. A passphrase is a special case of a password that is a sequence of words or other text. In this document, the use of the term "password" includes this special case.
Permutation	An ordered (re)arrangement of the elements of a (finite) set; a function that is both a one-to-one and onto mapping of a set to itself.
Plaintext data	In this Recommendation, data that will be encrypted by an encryption algorithm or obtained from ciphertext using a decryption algorithm.
Pre-shared key	A secret key that has been established between the parties who are authorized to use it by means of some secure method (e.g., using a secure manual-distribution process or automated key-establishment scheme).

---

<sup>6</sup> SP 800-38F: *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*.

Private key	<p>A cryptographic key used with an asymmetric-key (public-key) cryptographic algorithm that is not made public and is uniquely associated with an entity that is authorized to use it. In an asymmetric-key cryptosystem, the private key is associated with a public key. Depending on the algorithm that employs the private key, it may be used to:</p> <ol style="list-style-type: none"> <li>1. Compute the corresponding public key,</li> <li>2. Compute a digital signature that may be verified using the corresponding public key,</li> <li>3. Decrypt data that was encrypted using the corresponding public key, or</li> <li>4. Compute a key derivation key, which may then be used as an input to a key derivation process.</li> </ol>
Public key	<p>A cryptographic key used with an asymmetric-key (public-key) cryptographic algorithm that may be made public and is associated with a private key and an entity that is authorized to use that private key. Depending on the algorithm that employs the public key, it may be used to:</p> <ol style="list-style-type: none"> <li>1. Verify a digital signature that is signed by the corresponding private key,</li> <li>2. Encrypt data that can be decrypted by the corresponding private key, or</li> <li>3. Compute a piece of shared data (i.e., data that is known only by two or more specific entities).</li> </ol>
Public-key algorithm	See Asymmetric-key algorithm.
Random Bit Generator (RBG)	A device or algorithm that outputs bits that are computationally indistinguishable from bits that are independent and unbiased.
Rekey	A procedure in which a new cryptographic key is generated in a manner that is independent of the (old) cryptographic key that it will replace.
Secret key	A cryptographic key used by one or more (authorized) entities in a symmetric-key cryptographic algorithm; the key is not made public.
Secure channel	A path for transferring data between two entities or components that ensures confidentiality, integrity and replay protection, as well as mutual authentication between the entities or components. The secure channel may be provided using cryptographic, physical or procedural methods, or a combination thereof.

Security strength	A number associated with the amount of work (that is, the number of basic operations of some sort) required to break a cryptographic algorithm or system. Security strength is often expressed in bits. If the security strength is $S$ bits, then it is expected that (roughly) $2^S$ basic operations are required to break the algorithm or system.
<b>Shall</b>	This term is used to indicate a requirement of a Federal Information Processing Standard (FIPS) or a requirement that must be fulfilled to claim conformance to this Recommendation. Note that <b>shall</b> may be coupled with <b>not</b> to become <b>shall not</b> .
Shared secret	A secret value that has been computed during an execution of a key-establishment scheme between two parties, is known by both participants, and is used as input to a key-derivation method to produce keying material.
Support a security strength	A term applied to a method (e.g., an RBG, or a key with its associated cryptographic algorithm) that is capable of providing (at a minimum) the security strength required or desired for protecting data.
Symmetric key	See Secret key.
Symmetric-key algorithm	A cryptographic algorithm that uses the same secret key for its operation and, if applicable, for reversing the effects of the operation (e.g., an HMAC key for keyed hashing, or an AES key for encryption and decryption). Also known as a secret-key algorithm.
Target data	The data that is to be protected (e.g., a key or other sensitive data).
Trusted Party	A party that is trusted by its clients to generate cryptographic keys.

212 **2.2 Acronyms**

AES	Advanced Encryption Standard. See <a href="#">FIPS 197</a> . <sup>7</sup>
CMAC	Cipher-based MAC. See <a href="#">SP 800-38B</a> . <sup>8</sup>
CTR	Counter mode for a block cipher algorithm. See <a href="#">SP 800-38A</a> . <sup>9</sup>

<sup>7</sup> FIPS 197: *Advanced Encryption Standard*.<sup>8</sup> SP 800-38B: *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*.<sup>9</sup> SP 800-38A: *Recommendation for Block Cipher Modes of Operation - Methods and Techniques*.

DLC	Discrete Logarithm Cryptography.
FIPS	Federal Information Processing Standard.
HMAC	Keyed-Hash Message Authentication Code. See <a href="#">FIPS 198</a> . <sup>10</sup>
IFC	Integer Factorization Cryptography.
KDF	Key Derivation Function.
KMAC	KECCAK Message Authentication Code. See <a href="#">SP 800-185</a> . <sup>11</sup>
MAC	Message Authentication Code.
NIST	National Institute of Standards and Technology.
RBG	Random Bit Generator.
RSA	Rivest-Shamir-Adelman.
SP	Special Publication.

## 213 2.3 Symbols

Symbol	Meaning
$\oplus$	Bit-wise exclusive-or. A mathematical operation that is defined as: $0 \oplus 0 = 0,$ $0 \oplus 1 = 1,$ $1 \oplus 0 = 1, \text{ and}$ $1 \oplus 1 = 0.$
$\parallel$	Concatenation
$H(x)$	A cryptographic hash function with $x$ as an input.
$T(x, k)$	Truncation of the bit string $x$ to the leftmost $k$ bits of $x$ , where $k \leq$ the length of $x$ in bits.

## 214 3 General Discussion

### 215 3.1 Keys to Be Generated

216 This Recommendation addresses the generation of the cryptographic keys used in  
 217 cryptography. Key generation includes the generation of a key using the output of a random  
 218 bit generator, the derivation of a key from another key, the derivation of a key from a

<sup>10</sup> FIPS 198: *Keyed-Hash Message Authentication Code (HMAC)*.

<sup>11</sup> SP 800-185: *SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash*.

219 password, and key agreement performed by two entities using an **approved** key-agreement  
220 scheme. All keys **shall** be based directly or indirectly on the output of an **approved**  
221 Random Bit Generator (RBG). For the purposes of this Recommendation, keys that are  
222 derived during a key-agreement transaction (see [SP 800-56A](#) and [SP 800-56B](#)), derived  
223 from another key using a key derivation function (see [SP 800-108](#)) or derived from a  
224 password for storage applications (see [SP 800-132](#)<sup>12</sup> and [Section 6.5](#)) are considered to be  
225 indirectly generated from an RBG, since an ancestor key<sup>13</sup> or random value (e.g., the  
226 random value used to generate a key-agreement key pair) was obtained directly from the  
227 output of an **approved** RBG.

228 Two classes of cryptographic algorithms that require cryptographic keys have been  
229 **approved** for Federal government use: asymmetric-key algorithms and symmetric-key  
230 algorithms. The generation of keys for these algorithm classes is discussed in Sections [5](#)  
231 and [6](#).

### 232 **3.2 Where Keys are Generated**

233 Cryptographic keys **shall** be generated within [FIPS 140](#)<sup>14</sup>-compliant cryptographic  
234 modules. For explanatory purposes, consider the cryptographic module in which a key is  
235 generated to be the key-generating module. Any random value required by the key-  
236 generating module **shall** be generated within that module; that is, the RBG (or portion of  
237 the RBG<sup>15</sup>) that generates the random value **shall** be implemented within the FIPS 140  
238 cryptographic module that generates the key. The generated keys **shall** be transported  
239 (when necessary) using secure channels and **shall** be used by their associated cryptographic  
240 algorithm within FIPS 140-compliant cryptographic modules.

### 241 **3.3 Supporting a Security Strength**

242 A method (e.g., an RBG, or a key and its associated cryptographic algorithm) *supports a*  
243 *given security strength* if the security strength provided by that method is equal to or greater  
244 than the security strength required for protecting the target data; the actual security strength  
245 provided can be higher than required.

246 *Security strength supported by an RBG:* A well-designed RBG supports a given security  
247 strength if the amount of entropy (i.e., randomness) available in the RBG is equal to or  
248 greater than that security strength. The security strength supported depends on the secrecy  
249 of the information designated as the entropy bits and, when used for the generation of keys  
250 and other secret values, on the secrecy of the RBG output. For information regarding the  
251 security strength that can be supported by **approved** RBGs, see [SP 800-90A](#).<sup>16</sup>

252 *Security strength supported by an algorithm:* Discussions of cryptographic algorithms and

---

<sup>12</sup> SP 800-132: *Recommendation for Password-Based Key Derivation, Part 1: Storage Applications*.

<sup>13</sup> Ancestor key: A key that is used in the generation of another key. For example, an ancestor key for a key generated by a key derivation function would be the key-derivation key used by that key derivation function.

<sup>14</sup> “FIPS 140” without a reference to a particular version is intended to mean the version(s) of the FIPS that are currently being used for testing an implementation for conformance (e.g., FIPS 140-2).

<sup>15</sup> The RBG itself might be distributed (e.g., the entropy source may not co-reside with the algorithm that generates the (pseudo) random output).

<sup>16</sup> SP 800-90A: *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*.

253 the security strengths they can support, given certain choices of parameters and/or key  
254 lengths, are provided in [SP 800-57, Part 1](#).<sup>17</sup> The security strength of a cryptographic  
255 algorithm that uses keys of a certain size (i.e., length) is assessed under the assumption that  
256 those keys are generated using an **approved** process that provides entropy equal to or  
257 greater than the security strength assessed for that algorithm and key size (where both the  
258 entropy and the security strength are measured in bits).

259 *Security strength supported by a key:* The security strength that can be supported by a key  
260 depends on 1) the algorithm with which it is used, 2) the size of the key (see [SP 800-57,](#)  
261 [Part 1](#)), 3) the process that generated the key (e.g., the security strength supported by the  
262 RBG that was used to generate the key), and 4) how the key was handled (e.g., the security  
263 strength available in the method used to transport the key). The use of such terms as  
264 “security strength supported by a key” or “key supports a security strength” assumes that  
265 these factors have been taken into consideration. For example, if an **approved** RBG that  
266 supports a security strength of 128 bits has been used to generate a 128-bit key, and if  
267 (immediately after generation) the key is used with AES-128 to encrypt target data, then  
268 the key may be said to support a security strength of 128 bits in that encryption operation  
269 (for as long as the key is kept secret). However, if the 128-bit AES key is generated using  
270 an RBG that supports a security strength of only 112 bits, then the key can support a  
271 security strength of 112 bits, even though its length is still 128 bits; i.e., the security strength  
272 of the key has been reduced because of the process used for its generation (see item 3  
273 above).

## 274 **4 Using the Output of a Random Bit Generator**

275 Random bit strings required for the generation of cryptographic keys **shall** be obtained  
276 from the output of an **approved** Random Bit Generator (RBG); **approved** RBGs are  
277 specified in [SP 800-90](#).<sup>18</sup> The RBG **shall** be instantiated at a security strength that supports  
278 the security strength required to protect target data (i.e., the data that will be protected by  
279 the generated keys).

280 The output of an **approved** RBG may be used as specified in this section to obtain either  
281 a symmetric key or the random value needed to generate an asymmetric key pair.

282 Asymmetric key pairs require the use of an **approved** algorithm for their generation.  
283 Examples are those included in [FIPS 186](#)<sup>19</sup> for generating DSA, ECDSA, EdDSA and RSA  
284 keys. The generation of asymmetric key pairs from a random value is discussed in [Section](#)  
285 [5](#).

286 Methods for the generation of symmetric keys are discussed in [Section 6](#).

---

<sup>17</sup> SP 800-57, Part 1: *Recommendation for Key Management: General*.

<sup>18</sup> SP 800-90: *Recommendation for Random Number Generation*, consisting of SP 800-90A, SP 800-90B and SP 800-90C.

<sup>19</sup> FIPS 186: *Digital Signature Standard*.

287 Let  $K$  be either a symmetric key or the random value to be used as input to an **approved**  
288 asymmetric-key pair generation algorithm.  $K$  **shall** be a bit string value of the following  
289 form:

$$290 \qquad \qquad \qquad K = U \oplus V, \qquad \qquad \qquad (1)$$

291 where

- 292 •  $U$  is a bit string of the desired length that is obtained as the output of an **approved**  
293 Random Bit Generator (RBG) that is capable of supporting the desired security  
294 strength required to protect the target data,
- 295 •  $V$  is a bit string of the same length as  $U$ , and
- 296 • The value of  $V$  is determined in a manner that is independent of the value of  $U$  (and  
297 vice-versa).

298 The algorithm with which  $K$  will be used, and the security strength that this usage is  
299 intended to support will determine the required bit length and/or the security strength that  
300 this process must provide. Since there are no restrictions on the selection of  $V$  (other than  
301 its length and its independence from  $U$ ), a conservative approach necessitates an  
302 assumption that the process used to select  $U$  provides most (if not all) of the required  
303 entropy.

304 The independence requirement on  $U$  and  $V$  is interpreted in a computational and a statistical  
305 sense; that is, the computation of  $U$  does not depend on  $V$ , the computation of  $V$  does not  
306 depend on  $U$ , and knowing one of the values ( $U$  or  $V$ ) must yield no information that can  
307 be used to gain insight into the other value. Assuming that  $U$  is the output of an **approved**  
308 RBG, the following are examples of independently selected  $V$  values:

- 309 1.  $V$  is a constant (selected independently from the value of  $U$ ). (Note, that if  $V$  is a  
310 string of binary zeroes, then  $K = U$ , i.e., the output of an **approved** RBG.)
- 311 2.  $V$  is a key obtained using an **approved** key-derivation method from a key-  
312 derivation key and other input that is independent of  $U$ ; see [SP 800-108](#).
- 313 3.  $V$  is a key that was independently generated in another cryptographic module.  $V$   
314 was protected using an **approved** key-wrapping algorithm or transported using an  
315 **approved** key transport scheme during subsequent transport. Upon receipt, the  
316 protection on  $V$  is removed within the key-generating module that generated  $U$   
317 before combining  $V$  with  $U$ .
- 318 4.  $V$  is produced by hashing another bit string ( $V'$ ) using an **approved** hash function  
319 and (if necessary) truncating the result to the appropriate length before combining  
320 it with  $U$ . That is,  $V = T(H(V'), k)$  where  $T(x, k)$  denotes the truncation of bit string  
321  $x$  to its  $k$  leftmost bits, and  $k$  is the length of  $U$ . The bit string  $V'$  may be a)  
322 constant; b) a key derived from a shared secret during an **approved** key-agreement  
323 scheme between the key-generating module and another cryptographic module; or  
324 c) a key that was i) independently generated by another module, ii) sent using an  
325 **approved** key wrapping algorithm or transported using an **approved** key transport  
326 scheme, and iii) upon receipt, the protection on the key was removed.

## 327 **5 Generation of Key Pairs for Asymmetric-Key Algorithms**

328 Asymmetric-key algorithms, also known as public-key algorithms, require the use of  
329 asymmetric key pairs, consisting of a private key and a corresponding public key. A key  
330 pair can be used for the generation and verification of digital signatures (see [Section 5.1](#))  
331 or for key establishment (see [Section 5.2](#)). Each public/private key pair is associated with  
332 only one entity; this entity is known as the key-pair owner. The public key may be known  
333 by anyone, whereas the private key must be known and used only by the key-pair owner.  
334 Key pairs **shall** be generated by:

- 335 • The key-pair owner, or
- 336 • A Trusted Party that provides the key pair to the owner in a secure manner. The  
337 Trusted Party must be trusted by all parties that use the public key.

338 After key-pair generation, the key pair is retained by its owner; the public key is distributed  
339 to whomever needs to use it when interacting with the owner (see [Section 5.3](#)).

### 340 **5.1 Key Pairs for Digital Signature Schemes**

341 Digital signatures are generated on data to provide origin authentication, entity  
342 authentication, assurance of data integrity or support for signatory non-repudiation. Digital  
343 signatures are generated by a signer using a private key and verified by a receiver using a  
344 public key. The generation of key pairs for digital signature applications is addressed in  
345 [FIPS 186](#) for the DSA, RSA, ECDSA and EdDSA digital signature algorithms.

346 A value of  $K$ , computed as shown in [Section 4](#), **shall** be used to provide the random value  
347 needed to generate a key pair for the algorithms as specified in [FIPS 186](#). The maximum  
348 security strength that can be supported by these algorithms and the key lengths used by  
349 these algorithms are provided in [SP 800-57, Part 1](#).

350 For example, [SP 800-57, Part 1](#) states that a DSA key pair with a public-key length of 2048  
351 bits and a private-key length of 224 bits can support (at most) a security strength of 112  
352 bits. [FIPS 186](#) specifies that for such DSA key pairs, the random value used to determine  
353 a private key must be obtained using an RBG that supports a security strength of 112 bits.  
354 Using the method in [Section 4](#), a random value  $K$  that is to be used for the generation of  
355 the key pair is determined by  $U$  (a value of an appropriate bit length obtained from an RBG  
356 that supports a security strength of at least 112 bits) and  $V$  (which could be zero). The value  
357  $K$  is then used to determine the DSA key pair, as specified in [FIPS 186](#).

### 358 **5.2 Key Pairs for Key Establishment**

359 Key establishment includes both key agreement and key transport. Key agreement is a  
360 method of key establishment in which the resultant secret keying material is a function of  
361 information contributed by all participants in the key-establishment process (usually only  
362 two participants), so that no party can predetermine the value of the keying material  
363 independent of any other party's contribution. For key-transport, one party (the sender)  
364 selects a value for the secret keying material and then securely distributes that value to one  
365 or more other parties (the receiver).

366 **Approved** methods for generating the (asymmetric) key pairs used by **approved** key-  
367 establishment schemes between two parties are specified in [SP 800-56A](#) (for schemes that  
368 use finite-field or elliptic-curve cryptography) and in [SP 800-56B](#) (for schemes that use  
369 integer-factorization cryptography, e.g., RSA).

370 A value of  $K$ , computed as shown in [Section 4](#), **shall** be used to provide the random value<sup>20</sup>  
371 needed to generate key pairs for the finite field or elliptic curve schemes in [SP 800-56A](#),  
372 or to generate a key pair for the integer-factorization schemes specified in [SP 800-56B](#).  
373 The maximum security strength that can be supported by the **approved** key-establishment  
374 schemes and the key sizes used by these schemes is provided in [SP 800-57, Part 1](#).

### 375 **5.3 Distributing the Key Pairs**

376 A general discussion of the distribution of asymmetric key pairs is provided in [SP 800-57,](#)  
377 [Part 1](#).

378 The private key **shall** be kept secret. It **shall** either be generated 1) within the key-pair  
379 owner's cryptographic module (i.e., the key-pair owner's key-generating module), or 2)  
380 within the cryptographic module of an entity trusted by the key-pair owner and any relying  
381 party not to misuse the private key or reveal it to other entities (i.e., generated within the  
382 key-generating module of a Trusted Party, and securely transferred to the key-pair owner's  
383 cryptographic module).

384 If a private key is ever output from a cryptographic module, the key **shall** be output and  
385 transferred in a form and manner that provides appropriate assurance<sup>21</sup> of its confidentiality  
386 and integrity (e.g., using manual methods and multiparty control procedures or using  
387 automated key transport methods). The protection **shall** provide appropriate assurance that  
388 only the key-pair owner and/or the party that generated the key pair will be able to  
389 determine the value of the plaintext private key (e.g., the confidentiality and integrity  
390 protection for the private key uses a cryptographic mechanism that is at least as strong as  
391 the (maximum) security strength that must be supported by the asymmetric algorithm that  
392 will use the private key).

393 The public key of a key pair may be made public. However, it **shall** be distributed and  
394 verified in a manner that assures its integrity and association with the key-pair owner (e.g.,  
395 using a X.509 certificate that provides a level of cryptographic protection that is at least as  
396 strong as the security strength associated with the key pair).

### 397 **5.4 Key Pair Replacement**

398 Key pairs need to be replaced if the private key is compromised. Key pairs also need to be  
399 replaced from time to time to limit the amount of information that is protected by the key  
400 pair in case of a compromise of the private key (see Section 5.3 of [SP 800-57 Part 1](#));  
401 Section 5.3.4 of SP 800-57 Part 1 discusses the usage period for each key of the key pair

---

<sup>20</sup> Note that in Section 4, if  $V$  is all zeroes, then  $K$  (the random value) is the output of an RBG.

<sup>21</sup> The term "provide appropriate assurance" is used to allow various methods for the input and output of critical security parameters to/from the different security levels that may be implemented for a cryptographic module (see [FIPS 140](#) and Section 7.8 of [FIPS 140 IG](#)).

402 for both digital signature and key-establishment key pairs.

403 When asymmetric key pairs need to be replaced, they **shall** be generated and distributed as  
404 specified in Sections [5.1](#), [5.2](#) or [5.3](#), as appropriate.

## 405 **6 Generation of Keys for Symmetric-Key Algorithms**

406 Symmetric-key algorithms use the same (secret) key to both apply cryptographic protection  
407 to information<sup>22</sup> and to remove or verify the protection<sup>23</sup>. Keys used with symmetric-key  
408 algorithms must be known by only the entities authorized to apply, remove or verify the  
409 protection, and are commonly known as secret keys. A secret key is often known by  
410 multiple entities that are said to share or own the secret key, although it is not uncommon  
411 for a key to be generated, owned and used by a single entity (e.g., for secure storage). A  
412 secret key **shall** be generated by:

- 413 • One or more of the entities that will share the key, or
- 414 • A Trusted Party that provides the key to the intended sharing entities in a secure  
415 manner. The Trusted Party must be trusted by all entities that will share the key  
416 not to disclose the key to unauthorized parties or otherwise misuse the key (see [SP  
417 800-71](#)<sup>24</sup>).

418 A symmetric key  $K$  could be used, for example, to:

- 419 • Encrypt and decrypt data in an appropriate mode (e.g., using AES in the CTR mode  
420 as specified in [FIPS 197](#) and [SP 800-38A](#)),
- 421 • Generate Message Authentication Codes (e.g., using AES in the CMAC mode, as  
422 specified in [FIPS 197](#) and [SP 800-38B](#); HMAC, as specified in [FIPS 198](#); or  
423 KMAC, as specified in [SP 800-185](#)).
- 424 • Derive additional keys using a key-derivation function specified in [SP 800-108](#),  
425 where  $K$  is the pre-shared key that is used as the key-derivation key.

426 Section 6.1 discusses the generation of keys that are generated from the output of an RBG.  
427 [Section 6.2](#) discusses the distribution of a symmetric key after generation to the parties that  
428 need to use it. [Section 6.3](#) discusses a process called key agreement whereby keys are  
429 generated using a method that does not require explicit distribution of the generated key,  
430 since the key is already made available to the parties that need to use it during the key-  
431 generation process. Once two or more parties share an appropriate key, additional keys  
432 may be derived from that key without explicitly distributing those keys (see [Section 6.4](#)).

433 Sometimes symmetric keys are not intended to be shared, e.g., those keys intended to  
434 encrypt the key owner's data in storage. The preferred method for key generation is as

---

<sup>22</sup> For example, transform plaintext data into ciphertext data using an encryption operation or compute a message authentication code (MAC).

<sup>23</sup> For example, remove the protection by transforming the ciphertext data back to the original plaintext data using a decryption operation, or verify the protection by computing a message authentication code and comparing the newly computed MAC with a received MAC)

<sup>24</sup> SP 800-71: *Recommendation for Key Establishment Using Symmetric Block Ciphers*.

435 discussed in Section 6.1, but without distributing the key to other parties. Another method  
436 is to use a password known only by the data's owner to generate a storage key; however,  
437 this method is less secure than the previous methods (see [Section 6.5](#)).

438 A symmetric key can be combined with other keys or data to create another symmetric key  
439 to be used for cryptographic protection (see [Section 6.6](#)).

440 At some point, a symmetric key needs to be replaced for a number of possible reasons, e.g.,  
441 its cryptoperiod has been exceeded or it has been compromised (see [SP 800-57 Part 1](#));  
442 [Section 6.7](#) discusses key replacement.

## 443 **6.1 The "Direct Generation" of Symmetric Keys**

444 Symmetric keys that are to be directly generated from the output of an RBG **shall** be  
445 generated as specified in [Section 4](#), where  $K$  is the desired key. The length of the key to be  
446 generated is determined by the algorithm with which it will be used and the desired security  
447 strength to be provided; see [SP 800-57, Part 1](#) for discussions on key lengths and the  
448 (maximum) security strengths supported by symmetric-key algorithms.

## 449 **6.2 Distributing the Generated Symmetric Key**

450 The symmetric key generated within the key-generating module often needs to be shared  
451 with one or more other entities that have their own cryptographic modules. The key may  
452 be distributed manually or using an **approved** key transport or key wrapping method (see  
453 [SP 800-56B](#), [SP 800-38F](#) and [SP 800-71](#)). See [SP 800-57, Part 1](#) for further discussion.  
454 The method used for key transport or key wrapping **shall** support the desired security  
455 strength needed to protect the target data (i.e., the data to be protected using the symmetric  
456 key). The requirements for outputting a key from a cryptographic module are discussed in  
457 [FIPS 140](#).

## 458 **6.3 Symmetric Keys Generated Using Key-Agreement Schemes**

459 When an **approved** key-agreement scheme is available within an entity's key-generating  
460 module, a symmetric key may be established with another entity that has the same  
461 capability; this process results in a symmetric key that is shared between the two entities  
462 participating in the key-agreement transaction; further distribution of this symmetric key  
463 is not required between the two entities.

464 [SP 800-56A](#) and [SP 800-56B](#) provide several methods for pairwise key agreement.  
465 Asymmetric key-agreement keys are used with a key-agreement primitive to generate a  
466 shared secret. The shared secret is provided to a key-derivation method to derive keying  
467 material. [SP 800-56C](#)<sup>25</sup> specifies **approved** key-derivation methods for the key-agreement  
468 schemes in [SP 800-56A](#) and [SP 800-56B](#).

469 The maximum security strength that can be supported by a key derived in this manner is  
470 dependent on 1) the security strength supported by the asymmetric key pairs (as used  
471 during key establishment), 2) the key-derivation method used, 3) the length of the derived

---

<sup>25</sup> SP 800-56C: *Recommendation for Key-Derivation Methods in Key-Establishment Schemes*.

472 key and 4) the algorithm with which the derived key will be used. See [SP 800-57, Part 1](#).

#### 473 **6.4 Symmetric Keys Derived from a Pre-shared Key**

474 Symmetric keys are often derived using a key derivation function (KDF) and a pre-shared  
475 key known as a key-derivation key. The pre-shared key may have been:

- 476 • Generated from an **approved** RBG (see [Section 4](#)) and distributed as specified in  
477 [Section 6.2](#), if required,
- 478 • Agreed upon using a key-agreement scheme (see [Section 6.3](#)), or
- 479 • Derived using a KDF and a (different) pre-shared key as specified in [SP 800-108](#).

480 **Approved** methods for key derivation are provided in SP 800-108, which specifies  
481 **approved** KDFs for deriving keys from a pre-shared key (i.e., a key-derivation key). The  
482 KDFs are based on HMAC (as specified in [FIPS 198](#)) and CMAC (as specified in [SP 800-](#)  
483 [38B](#)).

484 If the derived keys need to be distributed to other entities, this may be accomplished as  
485 discussed in [Section 6.2](#).

486 In addition to the symmetric-key algorithm with which a derived key will be used, the  
487 security strength that can be supported by the derived key depends on the security strength  
488 supported by the key-derivation key and the KDF method used (see [SP 800-57, Part 1](#) for  
489 the maximum security strength that can be supported by HMAC and CMAC, and [SP 800-](#)  
490 [107](#)<sup>26</sup> for further discussions about the security strength of HMAC).

#### 491 **6.5 Symmetric Keys Derived from Passwords**

492 In a number of popular applications, keys are generated from passwords. This is a  
493 questionable practice, as the passwords usually contain very little entropy (i.e.,  
494 randomness), and are, therefore, easily guessed. However, **approved** methods for deriving  
495 keys from passwords for storage applications<sup>27</sup> are provided in [SP 800-132](#). For these  
496 applications, users are strongly advised to select passwords with a very large amount of  
497 entropy.

498 When a key is generated from a password, the entropy provided (and thus, the maximum  
499 security strength that can be supported by the generated key) **shall** be considered to be zero  
500 unless the password is generated using an **approved** RBG. In this case, the security  
501 strength that can be supported by the password (*password\_strength*) is no greater than the  
502 minimum of the security strength supported by the RBG (*RBG\_strength*) and the actual  
503 number of bits of RBG output (*RBG\_outlen*) used in the password. That is,  
504  $password\_strength \leq \min(RBG\_strength, RBG\_outlen)$ .

---

<sup>26</sup> SP 800-71: *Recommendation for Key Establishment Using Symmetric Block Ciphers*.

<sup>27</sup> For example, inside a FIPS 140-validated cryptographic module.

## 505 6.6 Symmetric Keys Produced by Combining Multiple Keys and Other 506 Data

507 When symmetric keys  $K_1, \dots, K_n$  are generated and/or established independently, they may  
508 be combined within a key-generating module to form a key  $K$ . Other items of data ( $V_1, \dots,$   
509  $V_m$ ) that are independent of these keys can also be combined with the  $K_i$  to form  $K$ , under  
510 the conditions specified below. Note that, while the  $K_i$  values are required to be secret, the  
511  $V_i$  values need not be kept secret.

512 The independent generation/establishment of the component keys  $K_1, \dots, K_n$  is interpreted  
513 in a computational and a statistical sense; that is, the computation of any particular  $K_i$  value  
514 does not depend on any one or more of the other  $K_i$  values, and it is not feasible to use  
515 knowledge of any proper subset of the  $K_i$  values to obtain any information about the  
516 remaining  $K_i$  values.

517 The independence of the component keys from the other items of data is also interpreted  
518 in a computational and a statistical sense. This means that the computation of the  $K_i$  values  
519 does not depend on any of the  $V_j$  values, the computation of the  $V_j$  values does not depend  
520 on any of the  $K_i$  values, and knowledge of the  $V_j$  values yields no information that can  
521 feasibly be used to gain insight into the  $K_i$  values. In cases where some (or all) of the  $V_j$   
522 values are secret, and the rest of the  $V_j$  values (if any) are public, “independence” also  
523 means that knowledge of the  $K_i$  values and public  $V_j$  values yields no information that can  
524 feasibly be used to gain insight into the secret  $V_j$  values.

525 The component symmetric keys  $K_1, \dots, K_n$  **shall** be generated and/or established  
526 independently using **approved** methods.<sup>28</sup> The other items of data (i.e.,  $V_1, \dots, V_m$ ) may  
527 be generated or obtained using any method that ensures their independence from those  
528 keys.

529 The **approved** methods for combining these symmetric keys (and other items of data) are:

- 530 1. Concatenating two or more keys, i.e.,  $K = K_1 \parallel \dots \parallel K_n$ . The sum of the bit lengths  
531 of the  $n$  component keys **shall** be equal to the required bit length of  $K$ .
- 532 2. Exclusive-Oring one or more symmetric keys, i.e.,  $K = K_1 \oplus \dots \oplus K_n$ . The length  
533 of each component key ( $K_i$ ) **shall** be equal to the length required for  $K$ .
- 534 3. Exclusive-Oring one or more symmetric keys and one or more other items of data,  
535 i.e.,  $K = K_1 \oplus \dots \oplus K_n \oplus V_1 \oplus \dots \oplus V_m$ . The length of each component key ( $K_i$ )  
536 and each item of data ( $V_i$ ) **shall** be equal to the length required for  $K$ .

537 Each  $K_i$  used in one of the three methods above **shall** be kept secret and **shall not** be used  
538 for any purpose other than the computation of a specific symmetric key  $K$  (i.e., a given  $K_i$   
539 **shall not** be used to generate more than one key).

## 540 6.7 Replacement of Symmetric Keys

541 Sometimes, a symmetric key may need to be replaced. This may be due to a compromise  
542 of the key or the end of the key’s cryptoperiod (see [SP 800-57, Part 1](#)). Replacement **shall**

---

<sup>28</sup> See Sections 6.1, 6.3 and 6.4.

543 be accomplished by a rekeying process. Rekeying is the replacement of a key with a new  
544 key that is generated independent of the value of the old key (i.e., knowledge of the old  
545 key provides no knowledge of the value of the replaced key and vice versa).

546 When a compromised key is replaced, the new key **shall** be generated in a manner that  
547 provides assurance of its independence from the compromised key. The new key may be  
548 generated using any method in [Section 5](#) with the following restrictions:

549 1. The method used **shall** provide assurance that there is no feasibly detectable  
550 relationship between the new key and the compromised key. To that end, the new  
551 key **shall not** be derived or updated using the compromised key.

552 2. If the compromised key was generated in a manner that depended (in whole or in  
553 part) on a password (see Sections [6.5](#) and [6.6](#)), then that password **shall** be changed  
554 prior to the generation of any new key; in particular, the new key(s) **shall** be  
555 generated in a manner that is independent of the old password value.

556 If an uncompromised symmetric key is to be replaced, it **shall** be replaced using any  
557 method in [Section 6](#) that supports the required amount of security strength. However, if the  
558 key to be replaced was generated in a manner that depended (in whole or in part) on a  
559 password (see Sections [6.5](#) and [6.6](#)), that password **shall** be changed prior to the generation  
560 of the new key.

561

**Appendix A: References**

- 562 [FIPS 140] National Institute of Standards and Technology (2002) *Security*  
563 *Requirements for Cryptographic Modules*. (U.S. Department of  
564 Commerce, Washington, D.C.), Federal Information Processing  
565 Standards Publication (FIPS) 140-2, May 25, 2001 (Change Notice 2,  
566 12/3/2002). <https://doi.org/10.6028/NIST.FIPS.140-2>
- 567 [FIPS 140 IG] National Institute of Standards and Technology, Canadian Centre for  
568 Cyber Security (2003) *Implementation Guidance for FIPS 140-2 and*  
569 *the Cryptographic Module Validation Program*, [Amended]. Available  
570 at [https://csrc.nist.gov/csrc/media/projects/cryptographic-module-](https://csrc.nist.gov/csrc/media/projects/cryptographic-module-validation-program/documents/fips140-2/FIPS1402IG.pdf)  
571 [validation-program/documents/fips140-2/FIPS1402IG.pdf](https://csrc.nist.gov/csrc/media/projects/cryptographic-module-validation-program/documents/fips140-2/FIPS1402IG.pdf)
- 572 [FIPS 180] National Institute of Standards and Technology (2015) *Secure Hash*  
573 *Standard (SHS)*. (U.S. Department of Commerce, Washington, D.C.),  
574 Federal Information Processing Standards Publication (FIPS) 180-4,  
575 August 2015. <https://doi.org/10.6028/NIST.FIPS.180-4>
- 576 [FIPS 186-4] National Institute of Standards and Technology (2013) *Digital*  
577 *Signature Standard (DSS)*. (U.S. Department of Commerce,  
578 Washington, D.C.), Federal Information Processing Standards  
579 Publication (FIPS) 186-4, July 2013.  
580 <https://doi.org/10.6028/NIST.FIPS.186-4>
- 581 [FIPS 197] National Institute of Standards and Technology (2001) *Advanced*  
582 *Encryption Standard (AES)*. (U.S. Department of Commerce,  
583 Washington, D.C.), Federal Information Processing Standards  
584 Publication (FIPS) 197, November 2001.  
585 <https://doi.org/10.6028/NIST.FIPS.197>
- 586 [FIPS 198] National Institute of Standards and Technology (2008) *The Keyed-Hash*  
587 *Message Authentication Code (HMAC)*. (U.S. Department of  
588 Commerce, Washington, D.C.), Federal Information Processing  
589 Standards Publication (FIPS) 198-1, July 2008.  
590 <https://doi.org/10.6028/NIST.FIPS.198-1>
- 591 [SP 800-38A] Dworkin MJ (2001) *Recommendation for Block Cipher Modes of*  
592 *Operation: Methods and Techniques*. (National Institute of Standards  
593 and Technology, Gaithersburg, Maryland), NIST Special Publication  
594 (SP) 800-38A, December 2001. [https://doi.org/10.6028/NIST.SP.800-](https://doi.org/10.6028/NIST.SP.800-38A)  
595 [38A](https://doi.org/10.6028/NIST.SP.800-38A)
- 596 [SP 800-38B] Dworkin MJ (2016) *Recommendation for Block Cipher Modes of*  
597 *Operation: the CMAC Mode for Authentication*. (National Institute of  
598 Standards and Technology, Gaithersburg, Maryland), NIST Special  
599 Publication (SP) 800-38B, May 2005 (includes updates as of  
600 10/06/2016). <https://doi.org/10.6028/NIST.SP.800-38B>

- 601 [SP 800-38F] Dworkin MJ (2012) *Recommendation for Block Cipher Modes of*  
602 *Operation: Methods for Key Wrapping*. (National Institute of Standards  
603 and Technology, Gaithersburg, Maryland), NIST Special Publication  
604 (SP) 800-38F, December 2012. [https://doi.org/10.6028/NIST.SP.800-](https://doi.org/10.6028/NIST.SP.800-38F)  
605 [38F](https://doi.org/10.6028/NIST.SP.800-38F)
- 606 [SP 800-56A] Barker EB, Chen L, Roginsky A, Vassilev A, Davis R (2018)  
607 *Recommendation for Pair-Wise Key-Establishment Schemes Using*  
608 *Discrete Logarithm Cryptography*. (National Institute of Standards and  
609 Technology, Gaithersburg, Maryland), NIST Special Publication (SP)  
610 800-56A, Rev. 3, April 2018. [https://doi.org/10.6028/NIST.SP.800-](https://doi.org/10.6028/NIST.SP.800-56Ar3)  
611 [56Ar3](https://doi.org/10.6028/NIST.SP.800-56Ar3)
- 612 [SP 800-56B] Barker EB, Chen L, Roginsky A, Vassilev A, Davis R, Simon S (2019)  
613 *Recommendation for Pair-Wise Key-Establishment Using Integer*  
614 *Factorization Cryptography*. (National Institute of Standards and  
615 Technology, Gaithersburg, Maryland), NIST Special Publication (SP)  
616 800-56B, Rev. 2, [Forthcoming].
- 617 [SP 800-56C] Barker EB, Chen L, Davis R (2018) *Recommendation for Key-*  
618 *Derivation Methods in Key-Establishment Schemes*. (National Institute  
619 of Standards and Technology, Gaithersburg, Maryland), NIST Special  
620 Publication (SP) 800-56C, Rev. 1, April 2018.  
621 <https://doi.org/10.6028/NIST.SP.800-56Cr1>
- 622 [SP 800-57-1] Barker EB (2016) *Recommendation for Key Management, Part 1:*  
623 *General*. (National Institute of Standards and Technology,  
624 Gaithersburg, Maryland), NIST Special Publication (SP) 800-57 Part 1,  
625 Rev. 4, January 2016. <https://doi.org/10.6028/NIST.SP.800-57pt1r4>
- 626 [SP 800-67] Barker EB, Mouha N (2017) *Recommendation for the Triple Data*  
627 *Encryption Algorithm (TDEA) Block Cipher*. (National Institute of  
628 Standards and Technology, Gaithersburg, Maryland), NIST Special  
629 Publication (SP) 800-67, Rev. 2, November 2017.  
630 <https://doi.org/10.6028/NIST.SP.800-67r2>
- 631 [SP 800-71] Barker EB, Barker WC (2018) *Recommendation for Key Establishment*  
632 *Using Symmetric Block Ciphers*. (National Institute of Standards and  
633 Technology, Gaithersburg, Maryland), Draft NIST Special Publication  
634 (SP) 800-71, July 2018. [https://csrc.nist.gov/publications/detail/sp/800-](https://csrc.nist.gov/publications/detail/sp/800-71/draft)  
635 [71/draft](https://csrc.nist.gov/publications/detail/sp/800-71/draft)
- 636 [SP 800-90A] Barker EB, Kelsey JM (2015) *Recommendation for Random Number*  
637 *Generation Using Deterministic Random Bit Generators*. (National  
638 Institute of Standards and Technology, Gaithersburg, Maryland), NIST  
639 Special Publication (SP) 800-90A, Rev. 1, June 2015.  
640 <https://doi.org/10.6028/NIST.SP.800-90Ar1>
- 641 [SP 800-90B] Sönmez Turan M, Barker EB, Kelsey JM, McKay KA, Baish ML, Boyle  
642 M (2018) *Recommendation for the Entropy Sources Used for Random*

- 643 *Bit Generation*. (National Institute of Standards and Technology,  
644 Gaithersburg, Maryland), NIST Special Publication (SP) 800-90B,  
645 January 2018. <https://doi.org/10.6028/NIST.SP.800-90B>
- 646 [SP 800-90C] Barker EB, Kelsey JM (2016) *Recommendation for Random Bit*  
647 *Generator (RBG) Constructions*. (National Institute of Standards and  
648 Technology, Gaithersburg, Maryland), Draft NIST Special Publication  
649 (SP) 800-90C, April 2016.  
650 <https://csrc.nist.gov/publications/detail/sp/800-90c/draft>
- 651 [SP 800-107] Dang QH (2012) *Recommendation for Applications Using Approved*  
652 *Hash Algorithms*. (National Institute of Standards and Technology,  
653 Gaithersburg, Maryland), NIST Special Publication (SP) 800-107, Rev.  
654 1, August 2012. <https://doi.org/10.6028/NIST.SP.800-107r1>
- 655 [SP 800-108] Chen L (2008) *Recommendation for Key Derivation Using*  
656 *Pseudorandom Functions (Revised)*. (National Institute of Standards  
657 and Technology, Gaithersburg, Maryland), NIST Special Publication  
658 (SP) 800-108, October 2009. <https://doi.org/10.6028/NIST.SP.800-108>
- 659 [SP 800-131A] Barker EB, Roginsky A (2019) *Transitioning the Use of Cryptographic*  
660 *Algorithms and Key Lengths*. (National Institute of Standards and  
661 Technology, Gaithersburg, Maryland), NIST Special Publication (SP)  
662 800-131A, Rev. 2, [Forthcoming].
- 663 [SP 800-132] Sönmez Turan M, Barker EB, Burr WE, Chen L (2010)  
664 *Recommendation for Password-Based Key Derivation, Part 1: Storage*  
665 *Applications*. (National Institute of Standards and Technology,  
666 Gaithersburg, Maryland), NIST Special Publication (SP) 800-132,  
667 December 2010.
- 668 [SP 800-135] Dang QH (2011) *Recommendation for Existing Application-Specific*  
669 *Key Derivation Functions*. (National Institute of Standards and  
670 Technology, Gaithersburg, Maryland), NIST Special Publication  
671 (SP) 800-135, Rev. 1, December 2011.  
672 <https://doi.org/10.6028/NIST.SP.800-135r1>
- 673 [SP 800-185] *Recommendation for Discrete Logarithm-based Cryptography:*  
674 *Elliptic Curve Domain Parameters*. (National Institute of Standards  
675 and Technology, Gaithersburg, Maryland), Draft NIST Special  
676 Publication (SP) 800-185, [Forthcoming].
- 677
- 678

679

## Appendix B: Revisions

680

A revision was made in 2018 with the following changes:

681

1. General: The Authority section (old Section 2) has been moved into the boilerplate (see page iii).

682

683

2. Section 2.1: Changes made to *cryptographic boundary*, *entropy*, *key-pair owner*, *key wrapping*, *rekey*, *shared secret* and *target data*.

684

685

Added: KMAC.

686

Removed *full-entropy*, *key update* and *non-repudiation*.

687

2. Section 2.2: Added KMAC and a reference to SP 800-185.

688

3. Section 3.3, para. 1, last line: Changed the reference to SP 800-90A instead of SP 800-90.

689

690

Last para.: The example has been expanded.

691

4. Section 4, para. 1, line 3: Removed the references to FIPS 186-2, X9.31 and X9.62, since the use of these RBGs is no longer allowed (see SP 800-131A).

692

693

Para. 3: Added EdDSA to the list of digital signature algorithms.

694

5. Section 5: A paragraph was inserted at the end to mention that the key pair needs to be distributed.

695

696

6. Section 5.1, para. 1, lines 1-2: Inserted entity authentication. Line 5: Added EdDSA.

697

7. Section 5.3, para. 3, lines 3-4: Inserted a parenthetical example.

698

8. Section 5.4: Added a new section on key replacement.

699

9. Section 6, bullet 2: Inserted a reference to SP 800-71. Bullet 4: Added KMAC, as specified in SP 800-185. Also added text introducing the remainder of Section 6.

700

701

10. Section 6.1, line 4: Inserted a reference to SP 800-71.

702

11. Section 6.3: Removed the figure and some of the associated text. Last paragraph: Removed the last four lines.

703

704

12. Section 6.4: Inserted a reference to SP 800-107.

705

13. Section 6.6: enlarged the subscripts for easier reading.

706

14. Section 6.7: The first paragraph was rewritten.

707

15. Appendix A: Updated the references.

708