

The attached DRAFT document (provided here for historical purposes) has been superseded by the following publication:

Publication Number: **NIST Interagency Report 8114**

Title: *Report on Lightweight Cryptography*

Publication Date: **March 2017**

- Final Publication: <https://doi.org/10.6028/NIST.IR.8114> (which links to <http://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf>).
- Information on other NIST cybersecurity publications and programs can be found at: <http://csrc.nist.gov/>

The following information was posted with the attached DRAFT document:

Aug 11, 2016

NISTIR 8114

DRAFT Report on Lightweight Cryptography

NIST-approved cryptographic standards were designed to perform well using general-purpose computers. In recent years, there has been increased deployment of small computing devices that have limited resources with which to implement cryptography. When current NIST-approved algorithms can be engineered to fit into the limited resources of constrained environments, their performance may not be acceptable. For these reasons, NIST started a lightweight cryptography project that was tasked with learning more about the issues and developing a strategy for the standardization of lightweight cryptographic algorithms. This report provides an overview of the lightweight cryptography project at NIST, and describes plans for the standardization of lightweight cryptographic algorithms.

Email comments to: lightweight-crypto@nist.gov(Subject: "Comments on Draft NISTIR 8114")

Comments due by: ***October 31, 2016***

DRAFT NISTIR 8114

Report on Lightweight Cryptography

Kerry A. McKay
Larry Bassham
Meltem Sönmez Turan
Nicky Mouha

DRAFT NISTIR 8114

Report on Lightweight Cryptography

Kerry A. McKay
Larry Bassham
Meltem Sönmez Turan
Nicky Mouha
*Computer Security Division
Information Technology Laboratory*

August 2016



U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

1 National Institute of Standards and Technology Internal Report 8114
2 29 pages (August 2016)

3
4
5 Certain commercial entities, equipment, or materials may be identified in this document in order to describe an
6 experimental procedure or concept adequately. Such identification is not intended to imply recommendation or
7 endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best
8 available for the purpose.

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

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

18
19 **Public comment period: August 11, 2016 through October 31, 2016**

20 National Institute of Standards and Technology
21 Attn: Computer Security Division, Information Technology Laboratory
22 100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
23 Email: lightweight-crypto@nist.gov

24 All comments are subject to release under the Freedom of Information Act (FOIA).
25

26

Reports on Computer Systems Technology

27 The Information Technology Laboratory (ITL) at the National Institute of Standards and
28 Technology (NIST) promotes the U.S. economy and public welfare by providing technical
29 leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test
30 methods, reference data, proof of concept implementations, and technical analyses to advance
31 the development and productive use of information technology. ITL's responsibilities include the
32 development of management, administrative, technical, and physical standards and guidelines for
33 the cost-effective security and privacy of other than national security-related information in
34 federal information systems.

35

36

Abstract

37 NIST-approved cryptographic standards were designed to perform well using general-purpose
38 computers. In recent years, there has been increased deployment of small computing devices that
39 have limited resources with which to implement cryptography. When current NIST-approved
40 algorithms can be engineered to fit into the limited resources of constrained environments, their
41 performance may not be acceptable. For these reasons, NIST started a lightweight cryptography
42 project that was tasked with learning more about the issues and developing a strategy for the
43 standardization of lightweight cryptographic algorithms. This report provides an overview of the
44 lightweight cryptography project at NIST, and describes plans for the standardization of
45 lightweight cryptographic algorithms.

46

47

Keywords

48 Constrained devices; lightweight cryptography; standardization

49

50

Acknowledgements

51 The authors would like to thank their NIST colleagues for providing valuable feedback during
52 the development of this publication.

53 **Executive Summary**

54 There are several emerging areas in which highly constrained devices are interconnected,
55 working in concert to accomplish some task. Examples of these areas include: automotive
56 systems, sensor networks, healthcare, distributed control systems, the Internet of Things (IoT),
57 cyber-physical systems, and the smart grid. Security and privacy can be very important in all of
58 these areas. Because the majority of modern cryptographic algorithms were designed for
59 desktop/server environments, many of these algorithms cannot be implemented in the
60 constrained devices used by these applications. When current NIST-approved algorithms can be
61 engineered to fit into the limited resources of constrained environments, their performance may
62 not be acceptable. For these reasons, NIST started a lightweight cryptography project that was
63 tasked with learning more about the issues and developing a strategy for the standardization of
64 lightweight cryptographic algorithms.

65 This report provides an overview of lightweight cryptography, summarizes the findings of the
66 NIST's lightweight cryptography project, and outlines NIST's plans for the standardization of
67 lightweight primitives. In particular, NIST has decided to create a portfolio of lightweight
68 primitives through an open process similar to the selection of block cipher modes of operation.
69 Algorithms will be recommended for use only in the context of profiles, which describe physical,
70 performance, and security characteristics. These profiles are intended to capture cryptographic
71 algorithm requirements imposed by devices and applications where lightweight cryptography is
72 needed. NIST will develop profiles based on community responses to questions, included in this
73 report, about application and device requirements for lightweight cryptography.

74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98

Table of Contents

- Executive Summary iii**
- 1 Introduction 1**
- 2 Overview of Lightweight Cryptography 2**
 - 2.1 Target Devices..... 2
 - 2.2 Performance Metrics..... 3
 - 2.2.1 Hardware-Specific Metrics.....3
 - 2.2.2 Software-Specific Metrics4
 - 2.3 Lightweight Cryptographic Primitives 4
 - 2.3.1 Lightweight Block Ciphers4
 - 2.3.2 Lightweight Hash Functions5
 - 2.3.3 Lightweight Message Authentication Codes.....6
 - 2.3.4 Lightweight Stream ciphers6
 - 2.4 NIST-Approved Cryptographic Primitives in Constrained Environments..... 6
 - 2.5 Lightweight Cryptography Standards 7
- 3 NIST’s Lightweight Cryptography Project 9**
 - 3.1 Scope and Design Considerations..... 9
 - 3.1.1 General Design Considerations.....9
 - 3.2 Profiles..... 11
 - 3.2.1 Profile Development11
 - 3.2.2 Profile Template and Sample Profiles 13
 - 3.3 Evaluation process..... 15
- References 17**

99 1 Introduction

100 The deployment of small computing devices such as RFID tags, industrial controllers, sensor
101 nodes and smart cards is becoming much more common. The shift from desktop computers to
102 small devices brings a wide range of new security and privacy concerns. It is challenging to
103 apply conventional standards to small devices. In many conventional cryptographic standards,
104 the tradeoff between security, performance and resource requirements was optimized for desktop
105 and server environments, and this makes them difficult or impossible to implement in resource-
106 constrained devices. When they can be implemented, their performance may not be acceptable.

107 Lightweight cryptography is a subfield of cryptography that aims to provide solutions tailored
108 for resource-constrained devices. There has been a significant amount of work done by the
109 academic community related to lightweight cryptography; this includes efficient
110 implementations of conventional cryptography standards, and the design and analysis of new
111 lightweight primitives and protocols.

112 In 2013, NIST initiated a lightweight cryptography project to study the performance of the
113 current NIST-approved cryptographic standards on constrained devices and to understand the
114 need for dedicated lightweight cryptography standards, and if the need is identified, to design a
115 transparent process for standardization. In 2015, NIST held the first Lightweight Cryptography
116 Workshop in Gaithersburg, MD, to get public feedback on the constraints and limitations of the
117 target devices, and requirements and characteristics of real-world applications of lightweight
118 cryptography.¹

119 Recently, NIST has decided to create a portfolio of lightweight primitives through an open
120 process similar to the selection of modes of operation of block ciphers [48]. In this report, we
121 aim to summarize the finding of the lightweight cryptography project and outline NIST's plans
122 for the standardization of lightweight primitives. This report also includes a list of questions to
123 the stakeholders of lightweight cryptography that will serve as the basis for determining
124 requirements. Responses to the questions should be sent to lightweight-crypto@nist.gov with the
125 subject line "Responses to questions on lightweight crypto requirements" before October 1,
126 2016.

127 The remainder of this report is organized as follows. Section 2 provides an overview of
128 lightweight cryptography, including metrics and developments. Section 3 provides information
129 about NIST's lightweight cryptography project, including the proposed path for the
130 standardization of lightweight algorithms, design considerations, and a profile template that will
131 be used in the evaluation process.

132

¹ For workshop presentations, visit http://www.nist.gov/itl/csd/ct/lwc_workshop2015.cfm.

133 2 Overview of Lightweight Cryptography

134 This section introduces various aspects of lightweight cryptography, including target devices,
135 performance metrics, applications and dedicated designs.

136 2.1 Target Devices

137 Lightweight cryptography targets a wide variety of devices that can be implemented on a broad
138 spectrum of hardware and software. On the high end of the device spectrum are servers and
139 desktop computers followed by tablets and smartphones. Conventional cryptographic algorithms
140 may perform well in these devices; therefore, these platforms may not require lightweight
141 algorithms. Finally, on the lower end of the spectrum are devices such as embedded systems,
142 RFID devices and sensor networks. Lightweight cryptography is primarily focused on the highly
143 constrained devices that can be found in the lower end of this spectrum.

Servers and Desktops	Conventional cryptography
Tablets and Smartphones	
Embedded Systems	Lightweight cryptography
RFID and Sensor Networks	

144 **Figure 1 Device Spectrum**

145 Microcontrollers are available with a wide array of performance attributes. Although 8-bit, 16-bit
146 and 32-bit microcontrollers are the most common, there are significant sales of 4-bit
147 microcontrollers for certain ultra-low cost applications. A wide variety of instruction sets exist,
148 typically only simple instructions are supported, and the number of instructions is often very
149 limited. This may result in a high number of cycles to execute common cryptographic
150 algorithms, which may make them too slow or energy-consuming for the intended application.
151 This is particularly a problem when it is necessary to satisfy real-time constraints using a limited
152 amount of energy.

153 For some microcontrollers, the amount of RAM and ROM can be extremely limited. For
154 example, the TI COP912C [62] has 64 bytes of RAM, and the NXP RS08 [52] can have as little
155 as 63 bytes of RAM. The Microchip PIC10/12/16 microcontrollers [45] exist in many variants
156 with 64 bytes of RAM and less, going down to as little as 16 bytes of RAM.

157 On the bottom of the spectrum there are RFID and sensor networks, which are often realized in
158 hardware (ASIC) in order to satisfy some of the most stringent implementation constraints. Of
159 particular interest are UHF RFID tags, for example using the widely deployed EPCGlobal Gen2
160 [22] and ISO/IEC 18000-63 [39] standards.

161 For RFID tags that are not battery-powered, only a limited amount of power is available from the
162 environment. Such devices require cryptographic algorithms that are not only implemented with
163 a very small amount of gate equivalents (GEs), but must meet stringent timing and power

164 requirements as well. A study of the constraints of such devices for cryptographic applications
165 was performed in [57].

166 Lightweight algorithms may be subject to various other constraints, a topic that will be explored
167 during the first phase of the standardization effort. The aforementioned examples are therefore
168 not intended to be exhaustive list, but to illustrate settings where conventional algorithms cannot
169 be implemented, in order to understand the need for lightweight alternatives.

170 2.2 Performance Metrics

171 In cryptographic algorithm design, there is a tradeoff between the performance and the resources
172 required for a given security level. Performance can be expressed in terms such as power and
173 energy consumption, latency, and throughput. The resources required for a hardware
174 implementation are usually summarized in gate area, gate equivalents, or slices. In software this
175 is reflected in register, RAM and ROM usage. Resource requirements are sometimes referred to
176 as costs, as adding more gates or memory tends to increase the production cost of a device.

177 Power and energy consumption are relevant metrics due to the nature of many constrained
178 devices. Power may be of particular importance in devices that harvest power from their
179 surroundings. An example would be an RFID chip that uses the electromagnetic field transmitted
180 by a reader to power its internal circuit. Energy consumption (i.e., power consumption over a
181 certain time period) is especially important in battery-operated devices that have a fixed amount
182 of stored energy. The batteries in some devices may be difficult or impossible to recharge or
183 replace once deployed. It should also be noted that power consumption depends on many factors,
184 such as the threshold voltage, the clock frequency and the technology used for implementation.

185 Latency is especially relevant for certain real-time applications, for example automotive
186 applications where very fast response times for components such as steering, airbags or brakes
187 are required. It can be defined as the measure of time between initial request of an operation and
188 producing the output. For example, the latency of an encryption operation is the time between
189 the initial request for encryption of a plaintext and the reply that returns the corresponding
190 ciphertext.

191 Throughput is the rate at which new outputs (e.g., authentication tags or ciphertext) are
192 produced. Unlike conventional primitives, high throughput may not be a design goal in
193 lightweight designs. However, moderate throughput is still required in most applications.

194 2.2.1 Hardware-Specific Metrics

195 Resource requirements for hardware platforms are typically described in terms of gate area. The
196 area of an implementation depends on the technology and the standard cell library, and is
197 measured in μm^2 . Area can be stated in terms of slices for FPGAs, or by *gate equivalents* (GEs)
198 for ASIC implementation.

199 On FPGAs, a slice is the basic reconfigurable unit, that contains a number of look-up tables
200 (LUTs), flip-flops and multiplexers. Slices are implemented differently on different FPGAs. The

201 number of LUTs, flip-flops and multiplexers depends on the FPGA family, as well as the number
202 of input and output bits of the LUTs.

203 For ASICs, one GE is equivalent to the area that is required by the two-input NAND gate. The
204 area in GE is obtained by dividing the area in μm^2 by the area of the NAND gate. The number of
205 GEs of a hardware implementation is therefore very specific to a particular technology, so that it
206 is not possible to directly compare the number of GEs of implementations across different
207 technologies.

208 A low-cost RFID tag may have a total gate count of 1,000-10,000 gates, out of which only 200-
209 2,000 may be used for security purposes [41]. Area requirements and power consumption can be
210 correlated, in which case minimizing area also tends to reduce the power consumption.

211 **2.2.2 Software-Specific Metrics**

212 For software applications, resource requirements can be measured by the number of registers, as
213 well as the number of bytes of RAM and ROM that are required. Functions that use a small
214 number of registers have a lower calling overhead, as fewer variables must be placed on the
215 stack before the registers can be overwritten. ROM is used to store the program code, and can
216 include fixed data, such as S-boxes or hardcoded round keys, while RAM is used to store
217 intermediate values that can be used in computations. This can lead to additional tradeoffs
218 between calculating values on the fly versus looking up values in a table.

219 **2.3 Lightweight Cryptographic Primitives**

220 Over the last decade, a number of lightweight crypto primitives (including block ciphers, hash
221 functions, message authentication codes and stream ciphers) have been proposed to bring
222 performance advantages over conventional cryptographic standards. These primitives differ from
223 conventional algorithms with the assumptions that lightweight primitives are not intended for a
224 wide range of applications, and may impose limits on the power of the attacker. For example, the
225 amount of data available to the attacker under a single key may be limited. However, it should be
226 noted that this does not mean that the lightweight algorithms are weak – rather the idea is to use
227 advancements to result in designs with a better balance between security, performance, and
228 resource requirements for specific resource-constrained environments.

229 **2.3.1 Lightweight Block Ciphers**

230 A number of lightweight block ciphers have been proposed to achieve performance advantages
231 over NIST's Advanced Encryption Standard (AES)[63], particularly AES-128. Some of these
232 ciphers were designed by simplifying conventional, well-analyzed block ciphers to improve their
233 efficiency. As an example, DESL [42] is a variant of DES, where the round function uses a
234 single S-box instead of eight and omits the initial and final permutations to improve the size of
235 the hardware implementation. Alternatively, some of the algorithms are dedicated block ciphers
236 that were designed from scratch. PRESENT [8] is one of the first lightweight block cipher
237 designs that was proposed for constrained hardware environments. SIMON and SPECK [6] are
238 families of lightweight block ciphers that were designed to be simple, flexible, and perform well

239 in hardware and software. There are also algorithms from 1990s such as RC5 [56], TEA [68] and
240 XTEA [51], which consist of simple round structures that make them suitable for constrained
241 software environments. A non-exhaustive list of lightweight block ciphers is provided in [67].

242 The performance benefits of lightweight block ciphers over conventional block ciphers are
243 achieved using lightweight design choices, such as:

244 - **Smaller block sizes:** To save memory, lightweight block ciphers may use smaller block
245 sizes than AES (e.g., 64 or 80 bits, rather than 128). It should also be noted that using
246 small block sizes reduces limits on the length of the plaintexts to be encrypted. For
247 example, outputs of a 64-bit block cipher can be distinguished from a random sequence
248 using around 2^{32} blocks for some of the approved modes of operations. Depending on the
249 algorithm, this may lead to plaintext recovery, key recovery or authentication tag
250 forgeries with non-negligible probabilities.

251 - **Smaller key sizes:** Some lightweight block ciphers use small key sizes (less than 96 bits)
252 for efficiency (e.g., 80-bit PRESENT). At the time of this writing, the minimum key size
253 required by NIST is 112 bits [4].

254 - **Simpler rounds:** The components and operations used in lightweight block ciphers are
255 typically simpler than those of conventional block ciphers. In lightweight designs using
256 S-boxes, 4-bit S-boxes are preferred over 8-bit S-boxes. This reduction in size results in
257 significant area savings. For example, the 4-bit S-box used in PRESENT required 28GEs,
258 whereas AES S-box required 395 GEs in [20]. For hardware-oriented designs, bit
259 permutations (such as those used in PRESENT), or recursive MDS matrices (as in
260 PHOTON [23] and LED [24]) may be preferred over complex linear layers. When rounds
261 are simpler, they may need to be iterated more times to achieve security.

262 - **Simpler key schedules:** Complex key schedules increase the memory, latency and the
263 power consumption of implementations; therefore, most of the lightweight block ciphers
264 use simple key schedules that can generate sub-keys on the fly. This may enable attacks
265 using related keys, weak keys, known keys or even chosen keys. When this is the case, it
266 is necessary to ensure that all keys are generated independently using a secure key
267 derivation function (KDF) [10, 11, 14, 60].

268 2.3.2 Lightweight Hash Functions

269 Conventional hash functions may not be suitable for constrained environments, mainly due to
270 their large internal state sizes and high power consumption requirements. This has led to the
271 development of lightweight hash functions, such as PHOTON [23], Quark [2] SPONGENT [7],
272 and Lesamnta-LW [26]. The expected usage of conventional and lightweight hash functions
273 differs in various aspects such as [54]:

274 - **Smaller internal state and output sizes:** Large output sizes are important for applications
275 that require collision resistance of hash functions. For applications that do not require
276 collision resistance, smaller internal and output sizes might be used. When a collision-

277 resistant hash function is required, it may be acceptable that this hash function has the
278 same security against preimage, second-preimage and collision attacks. This may reduce
279 the size of the internal state.

280 - *Smaller message size:* Conventional hash functions are expected to support inputs with
281 very large sizes (around 2^{64} bits). In most of the target protocols for lightweight hash
282 functions, typical input sizes are much smaller (e.g., at most 256 bits). Hash functions
283 that are optimized for short messages may therefore be more suitable for lightweight
284 applications.

285 2.3.3 Lightweight Message Authentication Codes

286 A message authentication code (MAC) generates a tag from a message and a secret key, which is
287 used to verify the authenticity of the message. Tag sizes are recommended to be at least 64 bits
288 for typical applications. For certain applications such as VoIP (Voice over IP), occasionally
289 accepting an inauthentic message may have limited impact on the security of the application, so
290 that shorter tags can be used after careful consideration. Chaskey [47], TuLP [21], and
291 LightMAC [43] are some of the examples of lightweight MAC algorithms.

292 2.3.4 Lightweight Stream ciphers

293 Stream ciphers are also promising primitives for constrained environments. The eSTREAM
294 competition [19], organized by the European Network of Excellence for Cryptology, aimed to
295 identify new stream ciphers that might be suitable for widespread adoption. The finalists of the
296 competition were announced in 2008 and included three stream ciphers for hardware applications
297 with restricted resources:

298 - *Grain* [25] is widely analyzed and provides implementation flexibility, and also has a
299 version that supports authentication.

300 - *Trivium* [15] is a widely analyzed, elegant and flexible design; however, it only supports
301 80-bit keys.

302 - *Mickey* [3] is less analyzed compared to Grain and Trivium, and its security mostly
303 depends on the hardness of analysis. It provides less implementation flexibility and
304 susceptible to timing and power analysis, due to irregular clocking.

305 2.4 NIST-Approved Cryptographic Primitives in Constrained Environments

306 This section discusses the performance of NIST-approved cryptographic standards in resource-
307 constrained environments.

- 308 - **Block ciphers:** There are two NIST-approved block cipher algorithms; AES and Triple
309 DES (TDEA)[5].² The AES family of block ciphers includes three variants AES-128,
310 AES-192, and AES-256 that support key sizes of 128, 192 and 256 bits, respectively. All
311 AES variants operate have a block size of 128 bits. For lightweight cryptography
312 purposes, the most suitable variant of the family is AES-128, due to the number of rounds
313 and size of the key schedule. Existing compact implementations of AES-128 require
314 2090 GEs [44] to 2400 GEs [46]. AES is mainly designed for software applications.
315 Using 8-bit AVR microcontrollers, encryption has been achieved in cycles per byte 124.6
316 and decryption in 181.3 cycles per byte, with a code size less than 2 Kbyte [53]. AES
317 performs very well on certain 8-bit microcontrollers, making it a good choice for those
318 platforms. Both encryption and decryption operations in block ciphers such as AES and
319 Triple-DES cannot be implemented on a Renesas RL78 16-bit microcontroller [55] when
320 the amount of ROM is limited to 512 bytes and RAM is limited to 128 bytes [13]. For
321 applications where the performance of AES is acceptable, AES should be used for
322 encryption.
- 323 - **Hash functions:** NIST-approved hash functions are specified in two FIPS standards:
324 FIPS 180-4 [65] specifies SHA-1,³ the SHA-2 family (namely, SHA-224, SHA-256,
325 SHA-384, SHA-512, SHA-512-224 and SHA-512/256) and FIPS 202 [66] specifies the
326 permutation-based SHA-3 family (namely, SHA3-224, SHA3-256, SHA3-384, and
327 SHA3-512). None of these approved hash functions are suitable for use in very
328 constrained environments, mainly due their large internal-state size requirements.
329 Ideguchi et al. [27] studied the RAM requirements of SHA-256, SHA-512 and various
330 SHA-3 candidates on low-cost 8-bit microcontrollers, and found that none of the NIST-
331 approved hash functions could be implemented within 64 bytes of RAM. The internal
332 state size for the SHA-3 family is mainly determined by the width of the underlying
333 1600-bit permutation. FIPS 202 additionally defines smaller-sized permutations with
334 sizes 25, 50, 100, 200, 400, and 800; some of these variants may later be used to define
335 lightweight variants of SHA-3, however currently these smaller variants are not approved
336 for use in hash functions.
- 337 - **Authenticated Encryption and MACs:** Authenticated encryption provides performance
338 and resource requirement advantages, because it simultaneously provides confidentiality
339 and integrity protection of messages. NIST approves the CCM [16] and GCM [18] block
340 cipher modes that provide authentication and encryption simultaneously. NIST also
341 approves standalone MACs, CMAC [17], GMAC [18], and HMAC [64], to be used for
342 generating and verifying message authentication.

343 2.5 Lightweight Cryptography Standards

344 ISO/IEC 29192, *Lightweight Cryptography*, is a six-part standard that specifies lightweight

² A third block cipher, Skipjack, is only approved for legacy-use decryption. See [SP800-131A] for more information.

³ SHA-1 is not approved for all common uses of a hash function. See [4] for further details.

345 cryptographic algorithms for confidentiality, authentication, identification, non-repudiation, and
346 key exchange. Part 1 [28] includes general information such as security, classification and
347 implementation requirements. Part 2 specifies the block ciphers PRESENT and CLEFIA [29].
348 Part 3 specifies the stream ciphers Enocoro and Trivium [ISO29192-3]. Part 4 specifies three
349 asymmetric techniques namely (i) identification scheme cryptoGPS, (ii) authentication and key
350 exchange mechanism ALIKE, and (iii) ID-based signature scheme IBS [30]. Part 5 specifies
351 three hash functions: PHOTON, SPONGENT, and Lesamnta-LW [40]. Part 6 is dedicated to
352 MACs and is currently under development.

353 ISO/IEC 29167, *Automatic Identification and Data Capture Techniques*, provides security
354 services for RFID air interface communications. Part 1 [31] describes the architecture, security
355 features, and requirements for security services for RFID devices. Crypto suites are defined in
356 additional parts. Currently, seven suites are published in [32-38]. Additional documents are
357 under development.

358 Cryptography Research and Evaluation Committees (CRYPTREC) is a project to evaluate and
359 monitor security of cryptographic techniques used in Japanese e-Government systems [12].
360 CRYPTREC publishes three types of cipher lists: e-Government Recommended Ciphers List,
361 Candidate Recommended Ciphers List and Monitored Ciphers List. The Lightweight
362 Cryptography working group of CRYPTREC, established in 2013, aims to study and support
363 appropriate lightweight cryptography solutions for e-government systems and any applications
364 where lightweight solutions are needed. The working group surveys research on state of the art in
365 lightweight cryptography and its applications, and performs implementation evaluations, and
366 published a report (in Japanese) [13] as a deliverable in 2015. The target algorithms for
367 implementation in the report were AES, Camellia [1], CLEFIA [59], PRESENT [8], LED [24],
368 Piccolo [58], TWINE [61], and PRINCE [9].

3 NIST's Lightweight Cryptography Project

370 NIST develops standards using several different approaches, as described in [50]. NIST has held
371 competitions to select the AES block cipher and the SHA-3 hash functions. These competitions
372 were significant efforts that took place over many years. For example, the SHA-3 competition
373 was announced in 2007, the winner was announced in 2012, and the standardization process was
374 concluded in 2015. Another approach is to adapt standards of other accredited standards
375 development organizations, as was done with HMAC and RSA standards. NIST researchers also
376 develop standards and guidelines in collaboration with experts in academia, industry and
377 government, if no suitable standard exists.

378 The landscape for lightweight cryptography is moving so quickly that a standard produced using
379 the competition model is likely to be outdated prior to standardization. Therefore, the most
380 suitable approach for lightweight cryptography, in terms of timeline and project goals, is to
381 develop new recommendations using an open call for proposals to standardize algorithms.

382 NIST is planning to develop and maintain a portfolio of lightweight primitives and modes that
383 are approved for limited use. Each algorithm in the portfolio will be tied to one or more *profiles*,
384 which consist of algorithm goals and acceptable ranges for metrics. This is in contrast to other
385 primitives and modes that are approved for general use. Any restrictions on use will be included
386 in the recommendation or standard where the primitives and modes of the portfolio are specified.
387 Algorithm transitions and deprecation guidance will be provided as algorithms in the portfolio
388 are phased out. The lightweight portfolio is not intended to offer alternative algorithms for
389 general use.

3.1 Scope and Design Considerations

391 The scope of NIST's lightweight cryptography project includes all cryptographic primitives and
392 modes that are needed in constrained environments. However, the initial focus of the project is
393 on block ciphers, hash functions, and message authentication codes. When long-term security is
394 needed, these algorithms should either aim for post-quantum security [49], or the application
395 should allow them to be easily replaceable by algorithms with post-quantum security.

396 While public key cryptography is not included in the initial focus, it is within the scope of this
397 project. However, it should be noted that public key schemes will only be considered for
398 inclusion in the portfolio under two conditions: 1) they are robust against quantum attacks, or 2)
399 use a combination of general public key cryptographic schemes with lightweight primitives (e.g.,
400 lightweight hash function). Protocol design is also an important part of achieving the desired
401 level of security while meeting requirements of a constrained environment, but is not within the
402 scope of this project.

3.1.1 General Design Considerations

404 While specific requirements vary by application, there are several generally desired properties
405 that NIST will be using to evaluate designs.

- 406 - **Security strength:** Any algorithm selected for the portfolio must provide adequate security.
407 More specifically, the security strength should be at least 112 bits.
- 408 - **Flexibility:** Efficient implementations of an algorithm should be possible across an
409 assortment of platforms. Algorithms should also allow a variety of implementations on a
410 single platform. Tunable algorithms, which use parameters to select properties such as state
411 size and key size, are desirable as they allow implementations with multiple options using
412 fewer resources than multiple algorithms that do not share logic, thereby supporting a wider
413 array of applications.
- 414 - **Low overhead for multiple functions:** Multiple functions (such as encryption and
415 decryption) that share the same core are preferred over functions that have completely
416 different logic. For example, a block cipher where the encryption and decryption operations
417 use similar round functions may be preferable over one that has distinct round functions for
418 encryption and decryption. Different primitives, such as a hash function and block cipher,
419 can also share logic, thus reducing the resources needed to implement multiple algorithms in
420 the same device.
- 421 - **Ciphertext expansion:** The size of the ciphertext has an impact on storage and transmission
422 costs. Algorithms and modes that do not significantly increase the amount of data are
423 desirable.
- 424 - **Side channel and fault attacks:** Implementations can leak sensitive information, particularly
425 information about the key or plaintext, in a variety of ways. Side channel attacks use
426 properties of the implementation during execution of the cryptographic operations, such
427 timing, power consumption, and electromagnetic emissions, to discover this sensitive
428 information. Fault attacks recover this sensitive information by introducing errors in the
429 computation. In the case of pervasive devices, this is particularly notable as attackers may
430 have physical access to the devices, and countermeasures for such attacks may not be present
431 due to constrained resources. Algorithms that are easy to protect against side channel and
432 fault attacks are desirable.
- 433 - **Limits on the number of plaintext-ciphertext pairs:** It may be permissible for algorithm
434 designers to assume an upper bound on the number of plaintext/ciphertext pairs, as this limit
435 can be justified for some applications by the constraints of the devices, (e.g., limitations on
436 the amount of data that are processed by the same key), or message formats defined by
437 protocols. However, it must be recognized that an attacker may mount attacks using plaintext
438 that was encrypted under multiple, independent keys (multi-key attacks), which are relevant
439 even when the amount of data encrypted under any single key is limited.
- 440 - **Related-key attacks:** These attacks allow an adversary to discover information about a key
441 by performing operations using multiple keys that, although unknown, have a known
442 relation. This is particularly a threat in protocols where keys are not chosen independent and
443 at random. Keys may be permanently burned into the hardware of constrained devices, with
444 no means of replacement. If this is the case, then related key attacks are only a practical
445 threat if an adversary can obtain several devices that have keys with a known relation. This

446 attack model still remains highly relevant for devices where the capability to update the key
 447 exists. The algorithms are expected to provide some resistance to related key attacks (e.g.,
 448 attacks require large number of related keys).

449 It may not be possible to satisfy all properties, in particular when this increases the resources
 450 beyond what is available for a given application. Still, any algorithm selected for the portfolio
 451 must provide adequate security. In particular, the security against key-recovery attacks should be
 452 at least 112 bits.

453 **3.2 Profiles**

454 NIST will evaluate and recommend algorithms based on profiles, which consist of a set of design
 455 goals, physical characteristics of target devices, performance characteristics imposed by the
 456 applications, and security characteristics.

457 Cryptographic primitives can be designed with a variety of goals in mind. The choices made in
 458 the design goals can affect the various characteristics. Initially, this project will focus on block
 459 ciphers, authenticated encryption schemes, hash functions, and message authentication codes.

460 Profiles should be designed to target classes of devices and applications – not necessarily
 461 specific applications. Profiles should be useful across a variety of applications. The
 462 characteristics that have been identified to be addressed in profiles are:

Physical characteristics	Performance characteristics	Security characteristics
Area (in GE)	Latency (in clock cycles)	Minimum security strength (bits)
Memory (RAM/ROM)	Throughput (cycles per byte)	Attack models
Implementation type (hardware, software, or both)	Power (μ W)	Side channel resistance requirements

463
 464 The appropriateness of an algorithm depends on the physical limitations of the device and the
 465 performance and security objectives imposed by the application.

466 **3.2.1 Profile Development**

467 When building profiles for lightweight cryptography, the numbers that express the physical,
 468 performance and security characteristics that apply to a specific constrained environment may
 469 not be meaningful by themselves. The reasoning behind them needs to be understood as well.

470 **Questions on Application and Device Requirements**

471 To develop profiles, NIST asks a series of questions to the stakeholders of lightweight

472 cryptography, in order to build relevant profiles for a variety of applications. This may help to
473 get a thorough understanding of a particular application and to identify the bottlenecks, or even
474 to identify additional constraints that are not immediately apparent. Responses to the questions
475 should be sent to lightweight-crypto@nist.gov with the subject line “Responses to questions on
476 lightweight crypto requirements” before October 1, 2016.

477 The list of questions is as follows. For a given application environment, not all questions may
478 apply.

- 479 1. What is the target application?
- 480 2. What types of functionality are required by the application (e.g., encryption,
481 authentication, hashing, signatures, etc.)?
- 482 3. Are any cryptographic algorithms currently used by the application? If so, which
483 algorithms? What motivated the choice for these algorithms? If not, why were certain
484 algorithms found to be unsuitable?
- 485 4. Are the algorithms mainly used locally (e.g., the direct communication between a tag and
486 a reader), or over a network?
- 487 5. Given the application, how difficult is it to replace a cryptographic algorithm?
- 488 6. Does the application mainly target hardware or software implementation, or are both
489 equally relevant? If so, why?
- 490 7. If software implementations are relevant, what platforms are considered (server, desktop,
491 laptop, smartphone, embedded, etc.)? Which specific types of processors (vendor and
492 architecture) are the main targets?
- 493 8. If hardware implementations are relevant, which types of hardware are considered
494 (FPGA, ASIC, etc.)? Which specific platforms are under consideration (vendor,
495 architecture, technology, standard-cell library, etc.)?
- 496 9. For software implementations, which resources are available for the cryptographic
497 computation? Are there limits on the amount of registers, RAM and ROM that are
498 available? If so, what technological or practical considerations can explain these limits?
- 499 10. For hardware implementations, are there limits on the amount of slices or GEs that are
500 available for the implementation? If so, what technological or practical considerations
501 can explain these limits?
- 502 11. Is the platform an inherently serial one, or can data be processed in parallel?
- 503 12. Is built-in support for cryptographic operations available on the platform? (Hardware
504 security modules, cryptographic instructions, cryptographically secure random or pseudo-
505 random bit generators?)
- 506 13. In the case of software implementations, is it necessary to obfuscate the implementation?
507 If so, why?
- 508 14. Is resistance against side-channel or fault attacks required? If no, why not?
- 509 15. Is some user-programmable non-volatile memory available?
- 510 16. How are keys generated? Where are they stored, and for how long?
- 511 17. How much data is processed under the same key? Are there inherent limitations to the
512 amount of data that is processed, e.g. resulting from the protocol or from technical
513 constraints?
- 514 18. Are the devices battery-powered, or do they draw their current from the environment?
515 What limits are imposed on the energy and/or power that is available to the device?

- 516 19. Does the device have to respond within a specific time? Is this a soft real-time (reduced
 517 usefulness after the deadline) or hard real-time (data becomes useless after deadline)
 518 requirement? How do these requirements translate to restrictions on any cryptographic
 519 algorithms that may be used in the application?
 520 20. What are typical sizes for a plaintext, ciphertext, message, authentication tag, etc.? What
 521 technological or practical factors determine their size? Would ciphertext expansion be
 522 acceptable, and if so by how many bytes?
 523 21. What are the concrete requirements for the security of the application? Which types of
 524 attacks are considered to be relevant, or irrelevant for the given application? Why so?
 525 22. Is there any other information that can be relevant to understand the application from a
 526 security or efficiency point of view?

527 **3.2.2 Profile Template and Sample Profiles**

528 It is not expected that one algorithm will necessarily meet all characteristics goals
 529 simultaneously. As such, profiles will be developed to support a set of characteristics and design
 530 goals. The proposed template is as follows:

Profile <profile name>	
Primitive	<i>Type of primitive</i>
Physical characteristics	<i>Name physical characteristic(s), and provide acceptable range(s) (e.g., 64 to 128 bytes of RAM)</i>
Performance characteristics	<i>Name performance characteristic(s), and provide acceptable range(s) (e.g., latency of no more than 5 ns)</i>
Security characteristics	<i>Minimum security strength, relevant attack models, side channel resistance requirements, etc.</i>
Design goals	<i>List design goals.</i>

531
 532 The following sample profiles are provided for example purposes only. Profiles for inclusion in
 533 the portfolio will be developed with the community in an open process. Sample applications are
 534 provided, but selected algorithms should be suitable for a variety of applications.

535 **Sample Profile #1**

536 The first sample profile is for a MAC algorithm having a low-area implementation in hardware,
 537 and is designed for short input messages.

538

Profile Sample_1	
Primitive	MAC
Physical characteristics	1600 to 1900 GEs, ASIC hardware implementation
Performance characteristics	Latency \leq 15 ns
Security characteristics	128-bit security, resistance to related key attacks, timing analysis
Design goals	Efficient for short input messages

539
540
541

A sample application using profile Sample_1 would be an RFID tag for asset tracking.

542 **Sample Profile #2**

543 The second sample profile is for an authenticated encryption algorithm with low latency. The
544 implementation may be in hardware or software, but should allow for decryption/verification.

Profile Sample_2	
Primitive	Block cipher
Physical characteristics	Hardware or software implementation
Performance characteristics	Latency \leq 20 ns
Security characteristics	128-bit security, resistance to power analysis
Design goals	Authenticated encryption

545

546 A sample application using profile Sample_2 would be command validation on a Controller Area
547 Network (CAN) bus.

548 **Sample Profile #3**

549 The third sample profile is for a MAC algorithm that uses minimal power.

Profile Sample_3	
Primitive	MAC
Physical characteristics	Hardware implementation
Performance characteristics	Power $\leq 10 \mu\text{W}$
Security characteristics	128-bit security, resistance against related key attacks, power analysis
Design goals	Resistance against tag forgeries

550 A sensor network node is an example of an application that may be compatible with profile
 551 Sample_3. Note that the same algorithm might be associated with both this profile and profile
 552 Sample_1.

553 3.3 Evaluation process

554 NIST will develop a submission and evaluation process for lightweight cryptographic algorithms
 555 that is similar to that of block cipher modes project [48]. There will be an open call for profiles
 556 and lightweight cryptographic algorithms. The submission requirements, guidelines, and sets of
 557 evaluation criteria will be made public on the Lightweight Cryptography project page
 558 (<http://www.nist.gov/itl/csd/ct/lwc-project.cfm>).

559 NIST will periodically hold workshops to discuss lightweight algorithms that are under
 560 consideration for the portfolio. These workshops will seek input from the community on
 561 cryptanalysis, implementations, and applications of the proposals.

562 The lwc-forum@nist.gov emailing list has been established for dialogue regarding NIST's
 563 Lightweight Cryptography project. To subscribe to the NIST lightweight cryptography mailing
 564 list, send an email message to lwc-forum-request@nist.gov, with a subject line "subscribe".

565 Tentative timeline:

- 566 - NIST solicits answers to the included list of questions about requirements from the
 567 community, based on current and upcoming application and device needs. Responses to
 568 the questions should be sent to lightweight-crypto@nist.gov with the subject line
 569 "Responses to questions on lightweight crypto requirements" before October 1, 2016.
 570 NIST will develop profiles based on the answers it receives, and these profiles will
 571 provide a starting point for discussion and call for primitives.
- 572 - NIST will hold the second Lightweight Cryptography Workshop on October 17-18, 2016.
 573 The purpose of this workshop will be to discuss this document, proposed profiles,
 574 comparison tools and methods, and recent work on cryptanalysis and implementations of
 575 lightweight cryptographic designs.
- 576 - NIST will publish a call for submissions of lightweight primitives in early 2017. The call
 577 will request submissions that are good solutions for the specified profiles.

- 578 - In late 2017, approximately six months after the call is published, NIST will begin
579 reviewing proposals.
- 580 - NIST will hold the third Lightweight Cryptography Workshop in early 2018 to discuss
581 proposals and plans for standardization.

582

583 **References**

- 584 [1] Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., and Tokita, T.,
585 *Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms — Design and Analysis*. Proc.
586 7th Annual International Workshop on Selected Areas in Cryptography (SAC 2000) Waterloo,
587 Ontario, Canada, August 14–15, 2000, pp. 39-56, http://dx.doi.org/10.1007/3-540-44983-3_4
- 588 [2] Aumasson, J.-P., Henzen, L., Meier, W., and Naya-Plasencia, M., *Quark: A Lightweight*
589 *Hash*, Journal of Cryptology, 2013, Vol. 26, (2), pp. 313-339, [http://dx.doi.org/10.1007/s00145-](http://dx.doi.org/10.1007/s00145-012-9125-6)
590 [012-9125-6](http://dx.doi.org/10.1007/s00145-012-9125-6)
- 591 [3] Babbage, S., and Dodd, M., *The MICKEY Stream Ciphers: ‘New Stream Cipher Designs*
592 *- The eSTREAM Finalists’* (Springer, 2008), LNCS 4986, pp. 191-209,
593 http://dx.doi.org/10.1007/978-3-540-68351-3_15
- 594 [4] Barker, E., and Roginsky, A., *Transitions: Recommendation for Transitioning the Use of*
595 *Cryptographic Algorithms and Key Lengths*, NIST Special Publication (SP) 800-131A Revision
596 1, National Institute of Standards and Technology, Gaithersburg, Maryland November 2015,
597 <http://dx.doi.org/10.6028/NIST.SP.800-131Ar1>
- 598 [5] Barker, W.C., and Barker, E., *Recommendation for the Triple Data Encryption Algorithm*
599 *(TDEA) Block Cipher*, NIST Special Publication (SP) 800-67 Revision 1, National Institute of
600 Standards and Technology, Gaithersburg, Maryland, January 2012,
601 <http://dx.doi.org/10.6028/NIST.SP.800-67r1>
- 602 [6] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L., *The*
603 *SIMON and SPECK Families of Lightweight Block Ciphers*, IACR Cryptology ePrint Archive,
604 2013, <http://eprint.iacr.org/2013/404>
- 605 [7] Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., and Verbauwhede, I.,
606 *SPONGENT: A Lightweight Hash Function*. Proc. 13th International Workshop on
607 Cryptographic Hardware and Embedded Systems (CHES 2011), Nara, Japan, September 28 –
608 October 1, 2011, LNCS 6917, pp. 312-325, http://dx.doi.org/10.1007/978-3-642-23951-9_21
- 609 [8] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B.,
610 Seurin, Y., and Vikkelsoe, C., *PRESENT: An Ultra-Lightweight Block Cipher*. Proc. 9th
611 International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2007),
612 Vienna, Austria, September 10-13, 2007, LNCS 4727, pp. 450-466,
613 http://dx.doi.org/10.1007/978-3-540-74735-2_31
- 614 [9] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R.,
615 Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., and Yalçın, T.,
616 *PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications*. Proc. 18th
617 International Conference on the Theory and Application of Cryptology and Information Security
618 (ASIACRYPT 2012), Beijing, China, December 2-6, 2012, pp. 208-225,
619 http://dx.doi.org/10.1007/978-3-642-34961-4_14

- 620 [10] Chen, L., *Recommendation for Key Derivation Using Pseudorandom Functions*
621 *(Revised)*, NIST Special Publication (SP) 800-108, National Institute of Standards and
622 Technology, Gaithersburg, Maryland, October 2009, <http://dx.doi.org/10.6028/NIST.SP.800-108>
- 623 [11] Chen, L., *Recommendation for Key Derivation through Extraction-then-Expansion*, NIST
624 Special Publication (SP) 800-56C, National Institute of Standards and Technology, Gaithersburg,
625 Maryland, November 2011, <http://dx.doi.org/10.6028/NIST.SP.800-56C>
- 626 [12] Cryptographic Research and Evaluation Committees, <http://www.cryptrec.go.jp/english/>
627 [accessed August 11, 2016]
- 628 [13] Cryptographic Research and Evaluation Committees, *CRYPTREC Report 2014*, Report of
629 the Cryptographic Technology Evaluation Committee, 296 pages, March 2015,
630 http://www.cryptrec.go.jp/report/c14_eval_web.pdf
- 631 [14] Dang, Q., *Recommendation for Existing Application-Specific Key Derivation Functions*,
632 NIST Special Publication (SP) 800-135 Revision 1, National Institute of Standards and
633 Technology, Gaithersburg, Maryland, December 2011, <http://dx.doi.org/10.6028/NIST.SP.800-135r1>
- 635 [15] De Cannière, C., and Preneel, B., *Trivium: 'New Stream Cipher Designs - The*
636 *eSTREAM Finalists'* (Springer, 2008), LNCS 4986, pp. 244-266, http://dx.doi.org/10.1007/978-3-540-68351-3_18
- 638 [16] Dworkin, M., *Recommendation for Block Cipher Modes of Operation: The CCM Mode*
639 *for Authentication and Confidentiality*, NIST Special Publication (SP) 800-38C, National
640 Institute of Standards and Technology, Gaithersburg, Maryland, May 2004,
641 <http://dx.doi.org/10.6028/NIST.SP.800-38C>
- 642 [17] Dworkin, M., *Recommendation for Block Cipher Modes of Operation: The CMAC Mode*
643 *for Authentication* NIST Special Publication (SP) 800-38B, National Institute of Standards and
644 Technology, Gaithersburg, Maryland, May 2005, <http://dx.doi.org/10.6028/NIST.SP.800-38B>
- 645 [18] Dworkin, M., *Recommendation for Block Cipher Modes of Operation: Galois/Counter*
646 *Mode (GCM) and GMAC*, NIST Special Publication (SP) 800-38D, National Institute of
647 Standards and Technology, Gaithersburg, Maryland, November 2007,
648 <http://dx.doi.org/10.6028/NIST.SP.800-38D>
- 649 [19] ECRYPT, *eSTREAM: the ECRYPT Stream Cipher Project*,
650 <http://www.ecrypt.eu.org/stream/>, [accessed August 10, 2016]
- 651 [20] Feldhofer, M., Dominikus, S., and Wolkerstorfer, J., *Strong Authentication for RFID*
652 *Systems Using the AES Algorithm*. Proc. 6th International Workshop on Cryptographic Hardware
653 and Embedded Systems (CHES 2004), Cambridge, MA, USA, August 11-13, 2004, pp. 357-
654 370, http://dx.doi.org/10.1007/978-3-540-28632-5_26
- 655 [21] Gong, Z., Hartel, P., Nikova, S., Tang, S.-H., and Zhu, B., *TuLP: A Family of*

- 656 *Lightweight Message Authentication Codes for Body Sensor Networks*, Journal of Computer
657 Science and Technology, 2014, Vol. 29, (1), pp. 53-68, [http://dx.doi.org/10.1007/s11390-013-](http://dx.doi.org/10.1007/s11390-013-1411-8)
658 [1411-8](http://dx.doi.org/10.1007/s11390-013-1411-8)
- 659 [22] GS1 EPCglobal Inc., EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID,
660 *Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960*
661 *MHz Version 2.0.1 Ratified*, 2015,
662 http://www.gs1.org/sites/default/files/docs/epc/Gen2_Protocol_Standard.pdf
- 663 [23] Guo, J., Peyrin, T., and Poschmann, A., *The PHOTON Family of Lightweight Hash*
664 *Functions*. Proc. 31st Annual International Cryptology Conference (CRYPTO 2011), Santa
665 Barbara, CA, USA, August 14-18, 2011, pp. 222-239, [http://dx.doi.org/10.1007/978-3-642-](http://dx.doi.org/10.1007/978-3-642-22792-9_13)
666 [22792-9_13](http://dx.doi.org/10.1007/978-3-642-22792-9_13)
- 667 [24] Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M., *The LED Block Cipher*. Proc. 13th
668 International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2011),
669 Nara, Japan, September 28 – October 1, 2011, pp. 326-341, [http://dx.doi.org/10.1007/978-3-](http://dx.doi.org/10.1007/978-3-642-23951-9_22)
670 [642-23951-9_22](http://dx.doi.org/10.1007/978-3-642-23951-9_22)
- 671 [25] Hell, M., Johansson, T., and Meier, W., *Grain: A Stream Cipher for Constrained*
672 *Environments*, International Journal of Wireless and Mobile Computing (IJWMC), 2007, Vol. 2,
673 (1), pp. 86-93, <http://dx.doi.org/10.1504/IJWMC.2007.013798>
- 674 [26] Hirose, S., Ideguchi, K., Kuwakado, H., Owada, T., Preneel, B., and Yoshida, H., *A*
675 *Lightweight 256-Bit Hash Function for Hardware and Low-End Devices: Lesamnta-LW*. Proc.
676 13th International Conference on Information Security and Cryptology (ICISC 2010), Seoul,
677 Korea, December 1-3, 2010, LNCS 6829, pp. 151-168, [http://dx.doi.org/10.1007/978-3-642-](http://dx.doi.org/10.1007/978-3-642-24209-0_10)
678 [24209-0_10](http://dx.doi.org/10.1007/978-3-642-24209-0_10)
- 679 [27] Ideguchi, K., Owada, T., and Yoshida, H., *A Study on RAM Requirements of Various*
680 *SHA-3 Candidates on Low-cost 8-bit CPUs*, IACR Cryptology ePrint Archive, 2009,
681 <http://eprint.iacr.org/2009/260>
- 682 [28] ISO, ISO/IEC 29192-1:2012, *Information Technology – Security Techniques –*
683 *Lightweight Cryptography – Part 1: General*, 2012,
684 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=56425
- 685 [29] ISO, ISO/IEC 29192-2:2012, *Information Technology – Security Techniques –*
686 *Lightweight Cryptography – Part 2: Block Ciphers*, 2012,
687 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=56552
- 688 [30] ISO, ISO/IEC 29192-4:2013, *Information Technology – Security Techniques –*
689 *Lightweight Cryptography – Part 4: Mechanisms Using Asymmetric Techniques*, 2013,
690 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=56427
- 691 [31] ISO, ISO/IEC 29167-1:2014, *Information Technology - Automated Identification and*
692 *Data Capture Techniques – Part 1: Security Services for RFID Air Interfaces*, 2014,

- 693 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61128
- 694 [32] ISO, ISO/IEC 29167-11:2014, *Information Technology - Automated Identification and*
695 *Data Capture Techniques – Part 11: Crypto Suite PRESENT-80 Security Services for Air*
696 *Interface Communications*, 2014,
697 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60441
- 698 [33] ISO, ISO/IEC 29167-10:2015, *Information Technology - Automated Identification and*
699 *Data Capture Techniques – Part 10: Crypto Suite AES-128 Security Services for Air Interface*
700 *Communications*, 2015,
701 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60440
- 702 [34] ISO, ISO/IEC 29167-12:2015, *Information Technology - Automated Identification and*
703 *Data Capture Techniques – Part 12: Crypto Suite ECC-DH Security Services for Air Interface*
704 *Communications*, 2015,
705 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60442
- 706 [35] ISO, ISO/IEC 29167-13:2015, *Information Technology - Automated Identification and*
707 *Data Capture Techniques – Part 13: Crypto Suite Grain-128A Security Services for Air Interface*
708 *Communications*, 2015,
709 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60682
- 710 [36] ISO, ISO/IEC 29167-14:2015, *Information Technology - Automated Identification and*
711 *Data Capture Techniques – Part 14: Crypto Suite AES OFB Security Services for Air Interface*
712 *Communications*, 2015,
713 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61130
- 714 [37] ISO, ISO/IEC 29167-16:2015, *Information Technology - Automated Identification and*
715 *Data Capture Techniques – Part 16: Crypto Suite ECDSA-ECDH Security Services for Air*
716 *Interface Communications*, 2015,
717 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61321
- 718 [38] ISO, ISO/IEC 29167-17:2015, *Information Technology - Automated Identification and*
719 *Data Capture Techniques – Part 17: Crypto Suite CryptoGPS Security Services for Air Interface*
720 *Communications*, 2015,
721 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61942
- 722 [39] ISO, ISO/IEC 18000-63:2015, *Information technology - Radio frequency identification*
723 *for item management - Part 63: Parameters for air interface communications at 860 MHz to 960*
724 *MHz Type C*, 2015,
725 http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63675
- 726 [40] ISO, ISO/IEC 29192-5:2016, *Information Technology – Security Techniques –*
727 *Lightweight Cryptography – Part 5: Hash-functions*, 2016,
728 http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=67173
- 729 [41] Juels, A., and Weis, S.A., *Authenticating Pervasive Devices with Human Protocols*. Proc.

- 730 25th Annual International Cryptology Conference (CRYPTO 2005), Santa Barbara, California,
731 USA, August 14-18, 2005, LNCS 3621, pp. 293-308, http://dx.doi.org/10.1007/11535218_18
- 732 [42] Leander, G., Paar, C., Poschmann, A., and Schramm, K., *New Lightweight DES Variants*.
733 Proc. 14th International Workshop on Fast Software Encryption (FSE 2007), Luxembourg,
734 Luxembourg, 2007, LNCS 4593, pp. 196-210, http://dx.doi.org/10.1007/978-3-540-74619-5_13
- 735 [43] Luykx, A., Preneel, B., Tischhauser, E., and Yasuda, K., *A MAC Mode for Lightweight*
736 *Block Ciphers*. Proc. 23rd International Conference on Fast Software Encryption (FSE 2016),
737 Bochum, Germany, March 20-23, 2016, LNCS 9783, pp. 43-59, http://dx.doi.org/10.1007/978-3-662-52993-5_3
738
- 739 [44] Mathew, S., Satpathy, S., Suresh, V., Anders, M., Kaul, H., Agarwal, A., Hsu, S., Chen,
740 G., and Krishnamurthy, R., *340 mV-1.1 V, 289 Gbps/W, 2090-Gate NanoAES Hardware*
741 *Accelerator With Area-Optimized Encrypt/Decrypt $GF(2^4)^2$ Polynomials in 22 nm Tri-Gate*
742 *CMOS*, IEEE Journal of Solid-State Circuits, 2015, Vol. 50, (4), pp. 1048-1058,
743 [http://ieeexplore.ieee.org/ielx7/4/7066864/07019004.pdf?tp=&arnumber=7019004&isnumber=7](http://ieeexplore.ieee.org/ielx7/4/7066864/07019004.pdf?tp=&arnumber=7019004&isnumber=7066864)
744 [066864](http://ieeexplore.ieee.org/ielx7/4/7066864/07019004.pdf?tp=&arnumber=7019004&isnumber=7066864)
- 745 [45] Microchip Technology Inc., *New/Popular 8-bit Microcontrollers Products*,
746 <http://www.microchip.com/ParamChartSearch/chart.aspx?branchID=1012> [accessed August 9,
747 2016]
- 748 [46] Moradi, A., Poschmann, A., Ling, S., Paar, C., and Wang, H., *Pushing the Limits: A Very*
749 *Compact and a Threshold Implementation of AES*. Proc. 30th Annual International Conference
750 on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2011), Tallinn,
751 Estonia, May 15-19, 2011, LNCS 6632, pp. 69-88, [http://dx.doi.org/10.1007/978-3-642-20465-](http://dx.doi.org/10.1007/978-3-642-20465-4_6)
752 [4_6](http://dx.doi.org/10.1007/978-3-642-20465-4_6)
- 753 [47] Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., and
754 Verbauwhede, I., *Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers*. Proc. 21st
755 International Conference on Selected Areas in Cryptography (SAC 2014), Montreal, QC,
756 Canada, August 14-15, 2014, pp. 306-323, http://dx.doi.org/10.1007/978-3-319-13051-4_19
- 757 [48] National Institute of Standards and Technology, *Block Cipher Modes*,
758 <http://csrc.nist.gov/groups/ST/toolkit/BCM/index.html> [accessed August 9, 2016]
- 759 [49] National Institute of Standards and Technology, *Post-Quantum Crypto Project*,
760 <http://csrc.nist.gov/groups/ST/post-quantum-crypto/index.html>, [accessed 2016, August 11]
- 761 [50] National Institute of Standards and Technology, *NIST Cryptographic Standards and*
762 *Guidelines Development Process*, NISTIR 7977, March 2016,
763 <http://dx.doi.org/10.6028/NIST.IR.7977>
- 764 [51] Needham, R.M., and Wheeler, D.J., *Tea extensions*, Technical Report, Computer
765 Laboratory, University of Cambridge, October 1997, <http://www.cix.co.uk/~klockstone/xtea.pdf>

- 766 [52] NXP, *8-bit RS08*, [http://www.nxp.com/products/microcontrollers-and-processors/more-](http://www.nxp.com/products/microcontrollers-and-processors/more-processors/8-16-bit-mcus/8-bit-rs08:RS08FAMILY)
767 [processors/8-16-bit-mcus/8-bit-rs08:RS08FAMILY](http://www.nxp.com/products/microcontrollers-and-processors/more-processors/8-16-bit-mcus/8-bit-rs08:RS08FAMILY) [accessed August 9, 2016]
- 768 [53] Osvik, D.A., Bos, J.W., Stefan, D., and Canright, D., *Fast Software AES Encryption*.
769 Proc. 17th International Workshop on Fast Software Encryption (FSE 2010), Seoul, Korea,
770 February 7-10, 2010, LNCS 6147, pp. 75-93, http://dx.doi.org/10.1007/978-3-642-13858-4_5
- 771 [54] Poschmann, A.Y.: *Lightweight Cryptography: Cryptographic Engineering for a*
772 *Pervasive World*. Ph.D. Thesis, Ruhr University Bochum, 2009, <http://d-nb.info/996578153>
- 773 [55] Renesas Electronics Corporation, *RL78 Family*, [https://www.renesas.com/en-](https://www.renesas.com/en-us/products/microcontrollers-microprocessors/rl78.html)
774 [us/products/microcontrollers-microprocessors/rl78.html](https://www.renesas.com/en-us/products/microcontrollers-microprocessors/rl78.html), [accessed August 11, 2016]
- 775 [56] Rivest, R.L., *The RC5 Encryption Algorithm*. Proc. Second International Workshop on
776 Fast Software Encryption (FSE 1994), Leuven, Belgium, December 14–16, 1994, LNCS 1008,
777 pp. 86-96, http://dx.doi.org/10.1007/3-540-60590-8_7
- 778 [57] Saarinen, M.-J.O., and Engels, D.W., *A Do-It-All-Cipher for RFID: Design Requirements*
779 *(Extended Abstract)*, IACR Cryptology ePrint Archive, 2012, <http://eprint.iacr.org/2012/317>
- 780 [58] Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., and Shirai, T., *Piccolo:*
781 *An Ultra-Lightweight Blockcipher*. Proc. 13th International Workshop on Cryptographic
782 Hardware and Embedded Systems (CHES 2011), Nara, Japan, September 28 – October 1, 2011,
783 pp. 342-357, http://dx.doi.org/10.1007/978-3-642-23951-9_23
- 784 [59] Shirai, T., Shibutani, K., Akishita, T., Moriai, S., and Iwata, T., *The 128-Bit Blockcipher*
785 *CLEFIA (Extended Abstract)*. Proc. 14th International Workshop on Fast Software Encryption
786 (FSE 2007), Luxembourg, Luxembourg, March 26-28, 2007, pp. 181-195,
787 http://dx.doi.org/10.1007/978-3-540-74619-5_12
- 788 [60] Sönmez Turan, M., Barker, E., Burr, W., and Chen, L., *Recommendation for Password-*
789 *Based Key Derivation: Part 1: Storage Applications*, NIST Special Publication (SP) 800-132,
790 National Institute of Standards and Technology, Gaithersburg, Maryland, December 2010,
791 <http://dx.doi.org/10.6028/NIST.SP.800-132>
- 792 [61] Suzuki, T., Minematsu, K., Morioka, S., and Kobayashi, E., *TWINE: A Lightweight Block*
793 *Cipher for Multiple Platforms*. Proc. 19th International Conference on Selected Areas in
794 Cryptography (SAC 2012), Windsor, ON, Canada, August 15-16 2012, pp. 339-354,
795 http://dx.doi.org/10.1007/978-3-642-35999-6_22
- 796 [62] Texas Instruments, *COP912C 8-Bit Microcontroller*,
797 <http://www.ti.com/product/COP912C> [accessed August 9, 2016]
- 798 [63] U.S. Department of Commerce, *Advanced Encryption Standard (AES)*, Federal
799 Information Processing Standards (FIPS) Publication 197, November 2001,
800 <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

- 801 [64] U.S. Department of Commerce, *The Keyed-Hash Message Authentication Code (HMAC)*,
802 Federal Information Processing Standards (FIPS) Publication 198-1, July 2008,
803 http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
- 804 [65] U.S. Department of Commerce, *Secure Hash Standard (SHS)* Federal Information
805 Processing Standards (FIPS) Publication 180-4, August 2015,
806 <http://dx.doi.org/10.6028/NIST.FIPS.180-4>
- 807 [66] U.S. Department of Commerce, *SHA-3 Standard: Permutation-Based Hash and*
808 *Extendable-Output Functions*, Federal Information Processing Standards (FIPS) Publication 202,
809 August 2015, <http://dx.doi.org/10.6028/NIST.FIPS.202>
- 810 [67] Université du Luxembourg, *Lightweight Block Ciphers*,
811 https://www.cryptolux.org/index.php/Lightweight_Block_Ciphers, [accessed May 10, 2016]
- 812 [68] Wheeler, D.J., and Needham, R.M., *TEA, A Tiny Encryption Algorithm*. Proc. Second
813 International Workshop on Fast Software Encryption (FSE 1994), Leuven, Belgium, December
814 14–16, 1994, LNCS 1008, pp. 363-366, http://dx.doi.org/10.1007/3-540-60590-8_29
- 815