NISTIR 8114

Report on Lightweight Cryptography

Kerry A. McKay
Larry Bassham
Meltem Sönmez Turan
Nicky Mouha

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8114

National Institute of
Standards and Technology
U.S. Department of Commerce

NISTIR 8114

Report on Lightweight Cryptography

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

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8114

March 2017

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

National Institute of Standards and Technology
Kent Rochford, Acting NIST Director and Under Secretary of Commerce for Standards and Technology

National Institute of Standards and Technology Internal Report 8114
21 pages (March 2017)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8114

Comments on this publication may be submitted to:

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

Reports on Computer Systems Technology

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

Abstract

NIST-approved cryptographic standards were designed to perform well on 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.

Keywords

Constrained devices; lightweight cryptography; standardization

Acknowledgements

The authors would like to thank their NIST colleagues, Lily Chen and Çağdaş Çalık for providing valuable feedback during the development of this publication.

Executive Summary

There are several emerging areas in which highly constrained devices are interconnected, working in concert to accomplish some task. Examples of these areas include: automotive systems, sensor networks, healthcare, distributed control systems, the Internet of Things (IoT), cyber-physical systems, and the smart grid. Security and privacy can be very important in all of these areas. Because the majority of modern cryptographic algorithms were designed for desktop/server environments, many of these algorithms cannot be implemented in the constrained devices used by these applications. 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 to investigate the issues and then develop a strategy for the standardization of lightweight cryptographic algorithms.

This report provides an overview of lightweight cryptography, summarizes the findings of NIST's lightweight cryptography project, and outlines NIST's plans for the standardization of lightweight algorithms. In particular, NIST has decided to create a portfolio of lightweight algorithms through an open process. This report includes a list of questions to the stakeholders of lightweight cryptography that will serve as the basis for determining requirements. NIST will develop profiles based on community responses to these questions. These profiles are intended to capture cryptographic algorithm requirements imposed by devices and applications where lightweight cryptography is needed. Algorithms will be recommended for use only in the context of profiles, which describe physical, performance, and security characteristics.

Table of Contents

1 Introduction

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

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

In 2013, NIST initiated a lightweight cryptography project to study the performance of the current NIST-approved cryptographic standards on constrained devices and to understand the need for dedicated lightweight cryptography standards, and if the need is identified, to design a transparent process for standardization. NIST held two Lightweight Cryptography Workshops in Gaithersburg, MD, to solicit public feedback on the constraints and limitations of the target devices, requirements and characteristics of real-world applications of lightweight cryptography.[1]

Recently, NIST has decided to create a portfolio of lightweight algorithms through an open process. In this report, we aim to summarize the finding of the lightweight cryptography project and to outline NIST's plans for the standardization of lightweight algorithms. This report also includes a list of questions to the stakeholders of lightweight cryptography that will serve as the basis for determining requirements. Responses to the questions should be sent to lightweight-crypto@nist.gov with the subject line "Responses to questions on lightweight crypto requirements".

The remainder of this report is organized as follows. Section 2 provides an overview of lightweight cryptography, including target devices, performance metrics and lightweight crypto algorithms, performance of NIST standards in constrained environments and lightweight crypto standards. Section 3 provides information about NIST's lightweight cryptography project, including the proposed path for the standardization of lightweight algorithms, design considerations, a list of questions for stakeholders and a profile template that will be used in the evaluation process.

[1] For workshop presentations, visit https://www.nist.gov/news-events/events/2015/07/lightweight-cryptography-workshop-2015, and https://www.nist.gov/news-events/events/2016/10/lightweight-cryptography-workshop-2016.

2 Overview of Lightweight Cryptography

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

2.1 Target Devices

Lightweight cryptography targets a wide variety of devices that can be implemented on a broad spectrum of hardware and software (see Figure 1). On the high end of the device spectrum are servers and desktop computers followed by tablets and smartphones. Conventional cryptographic algorithms generally perform well in these devices; therefore, these platforms do not require lightweight algorithms. On the lower end of the spectrum are devices such as embedded systems, RFID devices and sensor networks. Lightweight cryptography is primarily focused on the highly-constrained devices that can be found at this end of the spectrum.

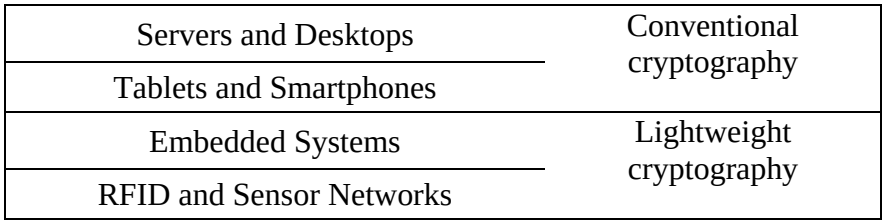

Figure 1: Device Spectrum

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

For some microcontrollers, the amount of random-access memory (RAM) and read-only memory (ROM) can be extremely limited. For example, the TI COP912C [66], NXP RS08 [56] or Microchip PIC10/12/16 microcontrollers [50] can have 64 bytes of RAM or less, going down to as little as 16 bytes of RAM.

On the bottom of the spectrum there are RFID and sensor networks, which are often realized in an application-specific integrated circuit (ASIC) in order to satisfy some of the most stringent implementation constraints. Of particular interest are ultra-high frequency (UHF) RFID tags, for example using the widely deployed EPCGlobal Gen2 [23] and ISO/IEC 18000-63 [38] standards.

For RFID tags that are not battery-powered, only a limited amount of power is available from the environment. Such devices require cryptographic algorithms that not only use a very small amount of gate equivalents (GEs), but also meet stringent timing and power requirements. A study on the constraints of such devices for cryptographic applications was performed in [61].

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

While lightweight cryptography primarily targets devices at the low end of the device spectrum, it is important to note that it may be necessary to implement lightweight algorithms at the high end of the spectrum as well. For example, many resource-constrained sensors may send information to an aggregator that, by most accounts, is not constrained. However, the aggregator must support lightweight algorithms in order to interoperate with the constrained sensors when they use lightweight cryptographic algorithms. In short, the environment and application need to be factored into the decision of whether or not conventional standards are acceptable. It is not just the limitation of a particular device that drives the need for lightweight cryptography, but also the other devices in the application that it directly interacts with.

2.2 Performance Metrics

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

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

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

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

2.2.1 Hardware-Specific Metrics

Resource requirements for hardware platforms are typically described in terms of gate area. The area of an implementation depends on the technology and the standard cell library, and is measured in μm^2. Area can be stated in terms of logic blocks for field-programmable gate arrays (FPGAs), or by GEs for ASIC implementations.

On FPGAs, a logic block is the basic reconfigurable unit, that contains a number of look-up tables (LUTs), flip-flops and multiplexers. Logic blocks are implemented differently on different FPGAs. The number of LUTs, flip-flops and multiplexers depends on the FPGA family, as well as the number of input and output bits of the LUTs.

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

A low-cost RFID tag may have a total gate count of 1,000 to 10,000 gates, out of which only 200 to 2,000 may be used for security purposes [46]. Area requirements and power consumption can be correlated, in which case minimizing area also tends to reduce the power consumption.

2.2.2 Software-Specific Metrics

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

2.3 Lightweight Cryptographic Primitives

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

2.3.1 Lightweight Block Ciphers

A number of lightweight block ciphers have been proposed to achieve performance advantages over NIST's Advanced Encryption Standard (AES)[67], particularly AES-128. Some of these ciphers were designed by simplifying conventional, well-analyzed block ciphers to improve their efficiency. As an example, DESL [47] is a variant of DES, where the round function uses a single S-box instead of eight and omits the initial and final permutations to improve the size of the hardware implementation. Alternatively, some of the algorithms are dedicated block ciphers that were designed from scratch. PRESENT [9] is one of the first lightweight block cipher designs that was proposed for constrained hardware environments. SIMON and SPECK [6] are families of lightweight block ciphers that were designed to be simple, flexible, and perform well in hardware and software. There are also algorithms from the 1990s such as RC5 [60], TEA [71] and XTEA [55], which consist of simple round structures that make them suitable for constrained software environments. A non-exhaustive list of lightweight block ciphers is provided in [7].

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

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

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

- ***Simpler rounds:*** The components and operations used in lightweight block ciphers are typically simpler than those of conventional block ciphers. In lightweight designs using S-boxes, 4-bit S-boxes are preferred over 8-bit S-boxes. This reduction in size results in significant area savings. For example, the 4-bit S-box used in PRESENT required 28 GEs, whereas the AES S-box required 395 GEs in [21]. For hardware-oriented designs, bit permutations (such as those used in PRESENT), or recursive MDS matrices (as in PHOTON [24] and LED [25]) may be preferred over complex linear layers. When rounds are simpler, they may need to be iterated more times to achieve security.

- ***Simpler key schedules:*** Complex key schedules increase the memory, latency and the power consumption of implementations; therefore, most of the lightweight block ciphers use simple key schedules that can generate sub-keys on the fly. This may enable attacks using related keys, weak keys, known keys or even chosen keys. Using a secure key derivation function (KDF) can prevent some of these attacks (for examples, see [11, 12, 15, 64]).

- ***Minimal implementations:*** There are several modes of operation and protocols that require only the encryption function of a block cipher. Some applications may require a device to only support one of the encryption or decryption operations. Implementing only the necessary functions of a cipher may require fewer resources than implementing the full cipher.

2.3.2 Lightweight Hash Functions

Conventional hash functions may not be suitable for constrained environments, mainly due to their large internal state sizes and high power consumption requirements. This has led to the development of lightweight hash functions, such as PHOTON [24], Quark [2], SPONGENT [8], and Lesamnta-LW [27]. The expected usage of conventional and lightweight hash functions differs in various aspects such as [58]:

- ***Smaller internal state and output sizes:*** Large output sizes are important for applications that require collision resistance of hash functions. For applications that do not require collision resistance, smaller internal states and output sizes might be used. When a collision-resistant hash function is required, it may be acceptable that this hash function has the same security against preimage, second-preimage and collision attacks. This may reduce the size of the internal state.

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

2.3.3 Lightweight Message Authentication Codes

A message authentication code (MAC) generates a tag from a message and a secret key, which is used to verify the authenticity and the integrity of the message. Tag sizes are recommended to be at least 64 bits for typical applications. For certain applications such as VoIP (Voice over IP), occasionally accepting an inauthentic message may have limited impact on the security of the application, so that shorter tags can be used after careful consideration. Chaskey [52], TuLP [22], and LightMAC [48] are some of the examples of lightweight MAC algorithms.

2.3.4 Lightweight Stream Ciphers

Stream ciphers are also promising primitives for constrained environments. The eSTREAM competition [20], organized by the European Network of Excellence for Cryptology, aimed to identify new stream ciphers that might be suitable for widespread adoption. The finalists of the competition were announced in 2008 and included three stream ciphers for hardware applications with restricted resources:

- Grain [26] is widely analyzed and provides implementation flexibility, and also has a version that supports authentication.

- Trivium [16] is a widely analyzed design; however, it only supports 80-bit keys.

- Mickey [3] is less analyzed compared to Grain and Trivium. It provides less implementation flexibility and is susceptible to timing and power analysis, due to irregular clocking.

2.4 NIST-Approved Cryptographic Primitives in Constrained Environments

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

- ***Block ciphers:*** There are two NIST-approved block cipher algorithms: AES and Triple Data Encryption Algorithm (TDEA)[5].[2] The AES family of block ciphers includes three variants AES-128, AES-192, and AES-256 that support key sizes of 128, 192 and 256 bits, respectively. All AES variants have a block size of 128 bits. For lightweight cryptography purposes, the most suitable variant of the family is AES-128, due to the number of rounds and the size of the key schedule. Existing compact implementations of AES-128 require 2090 GEs [49] to 2400 GEs [51]. AES is mainly designed for software applications. Using 8-bit AVR microcontrollers, encryption has been achieved in 124.6 cycles per byte and decryption in 181.3 cycles per byte, with a code size less than 2 Kbyte [57]. AES performs very well on certain 8-bit microcontrollers, making it a good choice for those platforms. However, it is not possible to implement the encryption and decryption functions of AES (or TDEA) simultaneously on a Renesas RL78 16-bit microcontroller [59] when the amount of ROM is limited to 512 bytes and RAM is limited to 128 bytes [14]. For applications where the performance of AES is acceptable, AES should be used.

- ***Hash functions:*** NIST-approved hash functions are specified in two Federal Information Processing Standards (FIPS) documents: FIPS 180-4 [69] specifies SHA-1[3] and the SHA-2 family (namely, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256) and FIPS 202 [70] specifies the permutation-based SHA-3 family (namely, SHA3-224, SHA3-256, SHA3-384, and SHA3-512). None of these approved hash functions are suitable for use in very constrained environments, mainly due their large internal-state size requirements. Ideguchi et al. [28] studied the RAM requirements of SHA-256, SHA-512 and various SHA-3 candidates on low-cost 8-bit microcontrollers, and found that none of the NIST-approved hash functions could be implemented within 64 bytes of RAM. The internal state size for the SHA-3 family is mainly determined by the width of the underlying 1600-bit permutation. FIPS 202 additionally defines smaller-sized permutations with 25, 50, 100, 200, 400, and 800 bits; some of these variants may later be used to define lightweight variants of SHA-3, however currently these smaller variants are not approved for use in hash functions.

[2] A third block cipher, Skipjack, is only approved for legacy-use decryption. See [4] for more information.

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

- ***Authenticated Encryption Algorithms and MACs:*** Authenticated encryption algorithms provide performance and resource requirement advantages, because they simultaneously provide confidentiality and integrity protection of messages. NIST approves the CCM [18] and GCM [17] block cipher modes that provide authentication and encryption simultaneously. NIST also approves standalone MACs, CMAC [19], GMAC [17], and HMAC [68], to be used for generating and verifying tags to provide message authentication.

2.5 Lightweight Cryptography Standards

ISO/IEC 29192, *Lightweight Cryptography*, is a six-part standard that specifies lightweight cryptographic algorithms for confidentiality, authentication, identification, non-repudiation, and key exchange. Part 1 includes general information such as security, classification and implementation requirements [39]. Part 2 specifies the block ciphers PRESENT and CLEFIA [40]. An amendment to Part 2 was proposed in 2014 to include the block ciphers SIMON and SPECK [6] with various block and key size combinations. In 2015, the first working drafts of the amendments with SIMON and SPECK were initiated. Part 3 specifies the stream ciphers Enocoro and Trivium [41]. Part 4 specifies three asymmetric techniques, namely (*i*) identification scheme cryptoGPS, (*ii*) authentication and key exchange mechanism ALIKE, and (*iii*) ID-based signature scheme IBS [42]. An amendment to Part 4 included an Elliptic Curve-based authentication scheme called ELLI [43]. Part 5 specifies three hash functions: PHOTON, SPONGENT, and Lesamnta-LW[44]. Part 6 is dedicated to MACs and is currently under development [45].

ISO/IEC 29167, *Automatic Identification and Data Capture Techniques*, provides security services for RFID air interface communications. Part 1 [29] describes the architecture, security features, and requirements for security services for RFID devices. Crypto suites are defined in additional parts. Currently, eight suites that specify the use of AES-128, PRESENT-80, ECC-DH, Grain-128A, AES OFB, ECDSA-ECDH, cryptoGPS, and RAMON security services for air interface communication are published in [30-37]. Additional documents are under development.

Cryptography Research and Evaluation Committees (CRYPTREC) is a project to evaluate and monitor security of cryptographic techniques used in Japanese e-Government systems [13]. CRYPTREC publishes three types of cipher lists: e-Government Recommended Ciphers List, Candidate Recommended Ciphers List and Monitored Ciphers List. The Lightweight Cryptography working group of CRYPTREC, established in 2013, aims to study and support appropriate lightweight cryptography solutions for e-government systems and any applications where lightweight solutions are needed. The working group surveys research on the state of the art in lightweight cryptography and its applications, performs implementation evaluations, and published a report (in Japanese) [14] as a deliverable in 2015. The target algorithms for implementation in the report were AES, Camellia [1], CLEFIA [63], PRESENT [9], LED [25], Piccolo [62], TWINE [65], and PRINCE [10].

3 NIST's Lightweight Cryptography Project

NIST develops standards using several different approaches, as described in [53]. NIST has held competitions to select the AES block cipher and the SHA-3 hash functions. These competitions were significant efforts that took place over many years. For example, the SHA-3 competition was announced in 2007, the winner was announced in 2012, and the standardization process was concluded in 2015. Another approach is to adapt standards of other accredited standards development organizations, as was done with HMAC and RSA standards. NIST researchers also develop standards and guidelines in collaboration with experts in academia, industry and government, if no suitable standard exists.

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

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

3.1 Scope

The scope of NIST's lightweight cryptography project includes all cryptographic primitives and modes that are needed in constrained environments. However, the initial focus of the project is on block ciphers, authenticated encryption schemes, hash functions, message authentication codes, cryptographic permutations, and stream ciphers. When long-term security is needed, these algorithms should either aim for post-quantum security [54], or the application should allow them to be easily replaceable by algorithms with post-quantum security.

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

3.2 Design Considerations

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

- ***Security strength*:** Any algorithm selected for the portfolio must provide adequate security. In general, the security strength should be at least 112 bits.

- ***Flexibility*:** Efficient implementations of an algorithm should be possible across an assortment of platforms. Algorithms should also allow a variety of implementations on a single platform. Tunable algorithms, which use parameters to select properties such as state size and key size, are desirable as they allow implementations with multiple options using fewer resources than multiple algorithms that do not share logic, thereby supporting a wider array of applications.

- ***Low overhead for multiple functions*:** Multiple functions (such as encryption and decryption) that share the same core are preferred over functions that have completely different logic. For example, a block cipher where the encryption and decryption operations use similar round functions may be preferable over one that has distinct round functions for encryption and decryption. Different primitives, such as a hash function and block cipher, can also share logic, thus reducing the resources needed to implement multiple algorithms in the same device.

- ***Ciphertext expansion*:** The size of the ciphertext has an impact on storage and transmission costs. Algorithms and modes that do not generate a ciphertext that is significantly longer than the plaintext are desirable.

- ***Side channel and fault attacks*:** Implementations can leak sensitive information, particularly information about the key or plaintext, in a variety of ways. Side channel attacks use properties of the implementation during execution of the cryptographic operations, such as timing, power consumption, and electromagnetic emissions, to discover this sensitive information. Fault attacks recover this sensitive information by introducing errors in the computation. In the case of pervasive devices, this is particularly notable as attackers may have physical access to the devices, and countermeasures for such attacks may not be present due to constrained resources. Algorithms that are easy to protect against side channel and fault attacks are desirable.

- ***Limits on the number of plaintext-ciphertext pairs*:** It may be permissible for algorithm designers to assume an upper bound on the number of plaintext/ciphertext pairs processed, as this limit can be justified for some applications by the constraints of the devices (e.g., limitations on the amount of data that are processed by the same key), or by message formats defined by protocols. However, it must be recognized that an attacker may mount attacks using plaintext that was encrypted under multiple, independent keys (multi-key attacks), which are relevant even when the amount of data encrypted under any single key is limited.

- ***Related-key attacks*:** These attacks allow an adversary to discover information about a key by performing operations using multiple unknown keys that have a known relation. This is particularly a threat in protocols where keys are not chosen independently and at random. Resistance to related key attacks may be desirable for some applications.

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

3.3 Profiles

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

Cryptographic algorithms can be designed with a variety of goals in mind. The choices made in the design goals can affect various characteristics.

Profiles will be designed to target classes of devices and applications – not necessarily specific applications. Profiles will be useful across a variety of applications. The characteristics that have been identified to be addressed in profiles are shown in the following table.

Table 1 Profile Characteristics

Physical characteristics	Performance characteristics	Security characteristics
Area (in GEs, logic blocks, or mm^2) Memory (RAM/ROM) Implementation type (hardware, software, or both) Energy (J)	Latency (in clock cycles or time period) Throughput (cycles per byte) Power (W)	Minimum security strength (bits) Attack models (e.g., related key, multi-key) Side channel resistance requirements

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

3.3.1 Profile Development

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

3.3.1.1 Questions on Application and Device Requirements

To develop profiles, NIST asks a series of questions to the stakeholders of lightweight cryptography, in order to build relevant profiles for a variety of applications. This may help to get a thorough understanding of a particular application and to identify the bottlenecks, or even to identify additional constraints that are not immediately apparent. Responses to the questions should be sent to lightweight-crypto@nist.gov with the subject line “Responses to questions on lightweight crypto requirements”.

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

1. What is the target application?
2. What types of functionality are required by the application (e.g., encryption, authentication, hashing, signatures, etc.)?
3. Are any cryptographic algorithms currently used by the application? If so, which algorithms? What motivated the choice for these algorithms? If not, why were certain algorithms found to be unsuitable?
4. Are the algorithms mainly used locally (e.g., the direct communication between a tag and a reader), or over a network?
5. Given the application, how difficult is it to replace a cryptographic algorithm?
6. Does the application mainly target hardware or software implementation, or are both equally relevant? If so, why?
7. If software implementations are relevant, what platforms are considered (server, desktop, laptop, smartphone, embedded, etc.)? Which specific types of processors (vendor and architecture) are the main targets?
8. If hardware implementations are relevant, which types of hardware are considered (FPGA, ASIC, etc.)? Which specific platforms are under consideration (vendor, architecture, technology, standard-cell library, etc.)?
9. For software implementations, which resources are available for the cryptographic computation? Are there limits on the amount of registers, RAM and ROM that are available? If so, what technological or practical considerations can explain these limits?
10. For hardware implementations, are there limits on the amount of logic blocks or GEs that are available for the implementation? If so, what technological or practical considerations can explain these limits?
11. Is the platform an inherently serial one, or can data be processed in parallel?
12. Is built-in support for cryptographic operations available on the platform? (Hardware security modules, cryptographic instructions, cryptographically secure random or pseudo-random bit generators?)
13. In the case of software implementations, is it necessary to obfuscate the implementation? If so, why?
14. Is resistance against side-channel or fault attacks required? If no, why not?
15. Is some user-programmable non-volatile memory available?
16. How are keys generated? Where are they stored, and for how long?
17. How much data is processed under the same key? Are there inherent limitations to the amount of data that is processed, e.g. resulting from the protocol or from technical constraints?
18. Are the devices battery-powered, or do they draw their current from the environment? What limits are imposed on the energy and/or power that is available to the device?
19. Does the device have to respond within a specific time? Is this a soft real-time (reduced usefulness after the deadline) or hard real-time (data becomes useless after deadline) requirement? How do these requirements translate to restrictions on any cryptographic algorithms that may be used in the application?
20. What are typical sizes for a plaintext, ciphertext, message, authentication tag, etc.? What technological or practical factors determine their size? Would ciphertext expansion be acceptable, and if so by how many bytes?

21. What are the concrete requirements for the security of the application? Which types of attacks are considered to be relevant, or irrelevant for the given application? Why so?
22. Is there any other information that can be relevant to understand the application from a security or efficiency point of view?

3.3.2 Profile Template

It is not expected that one algorithm will necessarily meet the needs of all applications simultaneously. As such, profiles will be developed to support a set of characteristics and design goals. In particular, the profile should capture the limiting factors of an application, as a manner of identifying where new lightweight algorithms can be used in lieu of NIST's conventional standards. A profile is not intended to be a full specification of requirements for a cryptographic implementation. The proposed template is given in Table 2.

Table 2 Proposed Template for Profiles

Profile <*profile name*>	
Functionality	*Purpose of cryptographic algorithm (e.g., encryption, authenticated encryption scheme, hashing, message authentication, etc.)*
Design goals	*List design goals.*
Physical characteristics	*Name physical characteristic(s), and provide acceptable range(s) (e.g., 64 to 128 bytes of RAM)*
Performance characteristics	*Name performance characteristic(s), and provide acceptable range(s) (e.g., latency of no more than 5 ns)*
Security characteristics	*Minimum security strength, relevant attack models, side channel resistance requirements, etc.*

Because the profile is only concerned with limitations, one or more of the characteristic fields, or the design goal field, may be blank.

3.4 Evaluation process

NIST will develop a submission and evaluation process for lightweight cryptographic algorithms. There will be an open call for profiles and lightweight cryptographic algorithms. The submission requirements, guidelines, and sets of evaluation criteria will be made public on the Lightweight Cryptography project page (http://www.nist.gov/itl/csd/ct/lwc-project.cfm).

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

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

This is an ongoing project that does not have a specified end date. As needs change, NIST will reevaluate the portfolio and add or deprecate portfolios and recommendations as necessary.

Tentative process:

- NIST solicits answers to the included list of questions about requirements from the community, based on current and upcoming application and device needs. Responses to the questions should be sent to lightweight-crypto@nist.gov with the subject line "Responses to questions on lightweight crypto requirements".
- NIST will develop profiles based on information from the community. Profiles will be announced online and on the lwc-forum mailing list. They will be subject to a public comment period of at least 30 days, and solicit feedback about whether current standards satisfy the profile.
- NIST will publish a call for submissions of lightweight cryptographic functions after a profile has been finalized. The call will request submissions that are good solutions for the specified profile(s).
- NIST will hold additional Lightweight Cryptography Workshops to discuss industry needs, profiles, proposals, and plans for standardization.

4 Summary

This report provided an overview of lightweight cryptography, and outlines NIST's plans on developing a portfolio of lightweight algorithms. The report included a series of questions to the stakeholders of lightweight cryptography, in order to build relevant profiles for a variety of applications. Based on community discussion and responses to the questions, NIST will develop profiles about application and device requirements for lightweight cryptography. Algorithms will be recommended for use only in the context of profiles, which describe physical, performance, and security characteristics.

References

[1] Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., and Tokita, T., *Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms — Design and Analysis*. Proc. 7th Annual International Workshop on Selected Areas in Cryptography (SAC 2000) Waterloo, Ontario, Canada, August 14–15, 2000, LNCS 2012, pp. 39-56, https://doi.org/10.1007/3-540-44983-3_4

[2] Aumasson, J.-P., Henzen, L., Meier, W., and Naya-Plasencia, M., *Quark: A Lightweight Hash*, Journal of Cryptology, 2013, Vol. 26, (2), pp. 313-339, https://doi.org/10.1007/s00145-012-9125-6

[3] Babbage, S., and Dodd, M., *The MICKEY Stream Ciphers*: 'New Stream Cipher Designs - The eSTREAM Finalists' (Springer, 2008), LNCS 4986, pp. 191-209, https://doi.org/10.1007/978-3-540-68351-3_15

[4] Barker, E., and Roginsky, A., *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, NIST Special Publication (SP) 800-131A Revision 1, National Institute of Standards and Technology, Gaithersburg, Maryland, November 2015, https://doi.org/10.6028/NIST.SP.800-131Ar1

[5] Barker, W.C., and Barker, E., *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, NIST Special Publication (SP) 800-67 Revision 1, National Institute of Standards and Technology, Gaithersburg, Maryland, January 2012, https://doi.org/10.6028/NIST.SP.800-67r1

[6] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L., *The SIMON and SPECK Families of Lightweight Block Ciphers*, IACR Cryptology ePrint Archive, 2013, http://eprint.iacr.org/2013/404

[7] Biryukov, A., and Perrin, L., *Lightweight Block Ciphers*, https://www.cryptolux.org/index.php/Lightweight_Block_Ciphers, [accessed May 10, 2016]

[8] Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., and Verbauwhede, I., *SPONGENT: A Lightweight Hash Function*. Proc. 13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2011), Nara, Japan, September 28 – October 1, 2011, LNCS 6917, pp. 312-325, https://doi.org/10.1007/978-3-642-23951-9_21

[9] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., and Vikkelsoe, C., *PRESENT: An Ultra-Lightweight Block Cipher*. Proc. 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2007), Vienna, Austria, September 10-13, 2007, LNCS 4727, pp. 450-466, https://doi.org/10.1007/978-3-540-74735-2_31

[10] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., and Yalçın, T.,

PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications. Proc. 18th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2012), Beijing, China, December 2-6, 2012, LNCS 7658, pp. 208-225, https://doi.org/10.1007/978-3-642-34961-4_14

[11] Chen, L., *Recommendation for Key Derivation through Extraction-then-Expansion*, NIST Special Publication (SP) 800-56C, National Institute of Standards and Technology, Gaithersburg, Maryland, November 2011, https://doi.org/10.6028/NIST.SP.800-56C

[12] Chen, L., *Recommendation for Key Derivation Using Pseudorandom Functions (Revised)*, NIST Special Publication (SP) 800-108, National Institute of Standards and Technology, Gaithersburg, Maryland, October 2009, https://doi.org/10.6028/NIST.SP.800-108

[13] Cryptographic Research and Evaluation Committees, http://www.cryptrec.go.jp/english/ [accessed August 11, 2016]

[14] Cryptographic Research and Evaluation Committees, *CRYPTREC Report 2014*, Report of the Cryptographic Technology Evaluation Committee, 296 pages, March 2015, http://www.cryptrec.go.jp/report/c14_eval_web.pdf

[15] Dang, Q., *Recommendation for Existing Application-Specific Key Derivation Functions*, NIST Special Publication (SP) 800-135 Revision 1, National Institute of Standards and Technology, Gaithersburg, Maryland, December 2011, https://doi.org/10.6028/NIST.SP.800-135r1

[16] De Cannière, C., and Preneel, B., *Trivium*: 'New Stream Cipher Designs - The eSTREAM Finalists' (Springer, 2008), LNCS 4986, pp. 244-266, https://doi.org/10.1007/978-3-540-68351-3_18

[17] Dworkin, M., *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, NIST Special Publication (SP) 800-38D, National Institute of Standards and Technology, Gaithersburg, Maryland, November 2007, https://doi.org/10.6028/NIST.SP.800-38D

[18] Dworkin, M., *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*, NIST Special Publication (SP) 800-38C, National Institute of Standards and Technology, Gaithersburg, Maryland, May 2004, https://doi.org/10.6028/NIST.SP.800-38C

[19] Dworkin, M., *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication* NIST Special Publication (SP) 800-38B, National Institute of Standards and Technology, Gaithersburg, Maryland, May 2005, https://doi.org/10.6028/NIST.SP.800-38B

[20] ECRYPT, *eSTREAM: the ECRYPT Stream Cipher Project*, http://www.ecrypt.eu.org/stream/, [accessed August 10, 2016]

[21] Feldhofer, M., Dominikus, S., and Wolkerstorfer, J., *Strong Authentication for RFID*

Systems Using the AES Algorithm. Proc. 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004), Cambridge, MA, USA, August 11-13, 2004, LNCS 3156, pp. 357-370, https://doi.org/10.1007/978-3-540-28632-5_26

[22] Gong, Z., Hartel, P., Nikova, S., Tang, S.-H., and Zhu, B., *TuLP: A Family of Lightweight Message Authentication Codes for Body Sensor Networks*, Journal of Computer Science and Technology, 2014, Vol. 29, (1), pp. 53-68, https://doi.org/10.1007/s11390-013-1411-8

[23] GS1 EPCglobal Inc., EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID, *Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHzVersion 2.0.1 Ratified*, 2015, http://www.gs1.org/sites/default/files/docs/epc/Gen2_Protocol_Standard.pdf

[24] Guo, J., Peyrin, T., and Poschmann, A., *The PHOTON Family of Lightweight Hash Functions*. Proc. 31st Annual International Cryptology Conference (CRYPTO 2011), Santa Barbara, CA, USA, August 14-18, 2011, LNCS 6841, pp. 222-239, https://doi.org/10.1007/978-3-642-22792-9_13

[25] Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M., *The LED Block Cipher*. Proc. 13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2011), Nara, Japan, September 28 – October 1, 2011, LNCS 6917, pp. 326-341, https://doi.org/10.1007/978-3-642-23951-9_22

[26] Hell, M., Johansson, T., and Meier, W., *Grain: A Stream Cipher for Constrained Environments*, International Journal of Wireless and Mobile Computing (IJWMC), 2007, Vol. 2, (1), pp. 86-93, https://doi.org/10.1504/IJWMC.2007.013798

[27] Hirose, S., Ideguchi, K., Kuwakado, H., Owada, T., Preneel, B., and Yoshida, H., *A Lightweight 256-Bit Hash Function for Hardware and Low-End Devices: Lesamnta-LW*. Proc. 13th International Conference on Information Security and Cryptology (ICISC 2010), Seoul, Korea, December 1-3, 2010, LNCS 6829, pp. 151-168, https://doi.org/10.1007/978-3-642-24209-0_10

[28] Ideguchi, K., Owada, T., and Yoshida, H., *A Study on RAM Requirements of Various SHA-3 Candidates on Low-cost 8-bit CPUs*, IACR Cryptology ePrint Archive, 2009, http://eprint.iacr.org/2009/260

[29] ISO, ISO/IEC 29167-1:2014, *Information Technology - Automated Identification and Data Capture Techniques - Part 1: Security Services for RFID Air Interfaces*, 2014, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61128

[30] ISO, ISO/IEC 29167-10:2015, *Information Technology - Automated Identification and Data Capture Techniques - Part 10: Crypto Suite AES-128 Security Services for Air Interface Communications*, 2015, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60440

[31] ISO, ISO/IEC 29167-11:2014, *Information Technology - Automated Identification and Data Capture Techniques - Part 11: Crypto Suite PRESENT-80 Security Services for Air Interface Communications*, 2014, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60441

[32] ISO, ISO/IEC 29167-12:2015, *Information Technology - Automated Identification and Data Capture Techniques - Part 12: Crypto Suite ECC-DH Security Services for Air Interface Communications*, 2015, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60442

[33] ISO, ISO/IEC 29167-13:2015, *Information Technology - Automated Identification and Data Capture Techniques - Part 13: Crypto Suite Grain-128A Security Services for Air Interface Communications*, 2015, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60682

[34] ISO, ISO/IEC 29167-14:2015, *Information Technology - Automated Identification and Data Capture Techniques - Part 14: Crypto Suite AES OFB Security Services for Air Interface Communications*, 2015, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61130

[35] ISO, ISO/IEC 29167-16:2015, *Information Technology - Automated Identification and Data Capture Techniques - Part 16: Crypto Suite ECDSA-ECDH Security Services for Air Interface Communications*, 2015, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61321

[36] ISO, ISO/IEC 29167-17:2015, *Information Technology - Automated Identification and Data Capture Techniques - Part 17: Crypto Suite CryptoGPS Security Services for Air Interface Communications*, 2015, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61942

[37] ISO, ISO/IEC 29167-19:2016, *Information Technology - Automated Identification and Data Capture Techniques - Part 19: Crypto suite RAMON security services for air interface communications*, 2016, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63176

[38] ISO, ISO/IEC 18000-63:2015, *Information Technology - Radio frequency identification for item management - Part 63: Parameters for air interface communications at 860 MHz to 960 MHz Type C*, 2015, http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63675

[39] ISO, ISO/IEC 29192-1:2012, *Information Technology - Security Techniques - Lightweight Cryptography - Part 1: General*, 2012, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=56425

[40] ISO, ISO/IEC 29192-2:2012, *Information Technology - Security Techniques - Lightweight Cryptography - Part 2: Block Ciphers*, 2012, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=56552

[41] ISO, ISO/IEC 29192-3:2012, *Information Technology - Security Techniques - Lightweight Cryptography - Part 3: Stream Ciphers*, 2012, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=56426

[42] ISO, ISO/IEC 29192-4:2013, *Information Technology - Security Techniques - Lightweight Cryptography - Part 4: Mechanisms Using Asymmetric Techniques*, 2013, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=56427

[43] ISO, ISO/IEC 29192-4:2013/Amd 1:2016, *Information Technology - Security Techniques - Lightweight Cryptography - Part 4: Mechanisms Using Asymmetric Techniques AMENDMENT 1*, 2016, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=64591

[44] ISO, ISO/IEC 29192-5:2016, *Information Technology - Security Techniques - Lightweight Cryptography - Part 5: Hash-functions*, 2016, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=67173

[45] ISO, ISO/IEC 29192-6, *Information Technology - Security Techniques - Lightweight Cryptography - Part 6: Message Authentication Codes (MACs)*, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=71116

[46] Juels, A., and Weis, S.A., *Authenticating Pervasive Devices with Human Protocols*. Proc. 25th Annual International Cryptology Conference (CRYPTO 2005), Santa Barbara, California, USA, August 14-18, 2005, LNCS 3621, pp. 293-308, https://doi.org/10.1007/11535218_18

[47] Leander, G., Paar, C., Poschmann, A., and Schramm, K., *New Lightweight DES Variants*. Proc. 14th International Workshop on Fast Software Encryption (FSE 2007), Luxembourg, Luxembourg, 2007, LNCS 4593, pp. 196-210, https://doi.org/10.1007/978-3-540-74619-5_13

[48] Luykx, A., Preneel, B., Tischhauser, E., and Yasuda, K., *A MAC Mode for Lightweight Block Ciphers*. Proc. 23rd International Conference on Fast Software Encryption (FSE 2016), Bochum, Germany, March 20-23, 2016, LNCS 9783, pp. 43-59, https://doi.org/10.1007/978-3-662-52993-5_3

[49] Mathew, S., Satpathy, S., Suresh, V., Anders, M., Kaul, H., Agarwal, A., Hsu, S., Chen, G., and Krishnamurthy, R., *340 mV-1.1 V, 289 Gbps/W, 2090-Gate NanoAES Hardware Accelerator With Area-Optimized Encrypt/Decrypt GF(2^4)^2 Polynomials in 22 nm Tri-Gate CMOS*, IEEE Journal of Solid-State Circuits, 2015, Vol. 50, (4), pp. 1048-1058, https://doi.org/10.1109/JSSC.2014.2384039

[50] Microchip Technology Inc., *New/Popular 8-bit Microcontrollers Products*, http://www.microchip.com/ParamChartSearch/chart.aspx?branchID=1012 [accessed August 9, 2016]

[51] Moradi, A., Poschmann, A., Ling, S., Paar, C., and Wang, H., *Pushing the Limits: A Very Compact and a Threshold Implementation of AES*. Proc. 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2011), Tallinn, Estonia, May 15-19, 2011, LNCS 6632, pp. 69-88, https://doi.org/10.1007/978-3-642-20465-4_6

[52] Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., and Verbauwhede, I., *Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers*. Proc. 21st International Conference on Selected Areas in Cryptography (SAC 2014), Montreal, QC, Canada, August 14-15, 2014, LNCS 8781, pp. 306-323, https://doi.org/10.1007/978-3-319-13051-4_19

[53] National Institute of Standards and Technology, *NIST Cryptographic Standards and Guidelines Development Process*, NISTIR 7977, March 2016, https://doi.org/10.6028/NIST.IR.7977

[54] National Institute of Standards and Technology, *Post-Quantum Crypto Project*, http://csrc.nist.gov/groups/ST/post-quantum-crypto/index.html, [accessed August 11, 2016]

[55] Needham, R.M., and Wheeler, D.J., *Tea extensions*, Technical Report, Computer Laboratory, University of Cambridge, October 1997, http://www.cix.co.uk/~klockstone/xtea.pdf

[56] NXP, *8-bit RS08*, http://www.nxp.com/products/microcontrollers-and-processors/more-processors/8-16-bit-mcus/8-bit-rs08:RS08FAMILY [accessed August 9, 2016]

[57] Osvik, D.A., Bos, J.W., Stefan, D., and Canright, D., *Fast Software AES Encryption*. Proc. 17th International Workshop on Fast Software Encryption (FSE 2010), Seoul, Korea, February 7-10, 2010, LNCS 6147, pp. 75-93, https://doi.org/10.1007/978-3-642-13858-4_5

[58] Poschmann, A.Y.: *Lightweight Cryptography: Cryptographic Engineering for a Pervasive World*. Ph.D. Thesis, Ruhr University Bochum, 2009, http://d-nb.info/996578153

[59] Renesas Electronics Corporation, *RL78 Family*, https://www.renesas.com/en-us/products/microcontrollers-microprocessors/rl78.html, [accessed August 11, 2016]

[60] Rivest, R.L., *The RC5 Encryption Algorithm*. Proc. Second International Workshop on Fast Software Encryption (FSE 1994), Leuven, Belgium, December 14–16, 1994, LNCS 1008, pp. 86-96, https://doi.org/10.1007/3-540-60590-8_7

[61] Saarinen, M.-J.O., and Engels, D.W., *A Do-It-All-Cipher for RFID: Design Requirements (Extended Abstract)*, IACR Cryptology ePrint Archive, 2012, http://eprint.iacr.org/2012/317

[62] Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., and Shirai, T., *Piccolo: An Ultra-Lightweight Blockcipher*. Proc. 13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2011), Nara, Japan, September 28 – October 1, 2011, LNCS 6917, pp. 342-357, https://doi.org/10.1007/978-3-642-23951-9_23

[63] Shirai, T., Shibutani, K., Akishita, T., Moriai, S., and Iwata, T., *The 128-Bit Blockcipher CLEFIA (Extended Abstract)*. Proc. 14th International Workshop on Fast Software Encryption (FSE 2007), Luxembourg, Luxembourg, March 26-28, 2007, LNCS 4593, pp. 181-195, https://doi.org/10.1007/978-3-540-74619-5_12

[64] Sönmez Turan, M., Barker, E., Burr, W., and Chen, L., *Recommendation for Password-*

Based Key Derivation: Part 1: Storage Applications, NIST Special Publication (SP) 800-132, National Institute of Standards and Technology, Gaithersburg, Maryland, December 2010, https://doi.org/10.6028/NIST.SP.800-132

[65] Suzaki, T., Minematsu, K., Morioka, S., and Kobayashi, E., *TWINE: A Lightweight Block Cipher for Multiple Platforms*. Proc. 19th International Conference on Selected Areas in Cryptography (SAC 2012), Windsor, ON, Canada, August 15-16 2012, LNCS 7707, pp. 339-354, https://doi.org/10.1007/978-3-642-35999-6_22

[66] Texas Instruments, *COP912C 8-Bit Microcontroller*, http://www.ti.com/product/COP912C [accessed August 9, 2016]

[67] U.S. Department of Commerce, *Advanced Encryption Standard (AES)*, Federal Information Processing Standards (FIPS) Publication 197, November 2001, https://doi.org/10.6028/NIST.FIPS.197

[68] U.S. Department of Commerce, *The Keyed-Hash Message Authentication Code (HMAC)*, Federal Information Processing Standards (FIPS) Publication 198-1, July 2008, https://doi.org/10.6028/NIST.FIPS.198-1

[69] U.S. Department of Commerce, *Secure Hash Standard (SHS)* Federal Information Processing Standards (FIPS) Publication 180-4, August 2015, https://doi.org/10.6028/NIST.FIPS.180-4

[70] U.S. Department of Commerce, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, Federal Information Processing Standards (FIPS) Publication 202, August 2015, https://doi.org/10.6028/NIST.FIPS.202

[71] Wheeler, D.J., and Needham, R.M., *TEA, A Tiny Encryption Algorithm*. Proc. Second International Workshop on Fast Software Encryption (FSE 1994), Leuven, Belgium, December 14–16, 1994, LNCS 1008, pp. 363-366, https://doi.org/10.1007/3-540-60590-8_29