Public Comments Received on
NISTIR 8114: Draft Report on Lightweight Cryptography
(Comment Period Closed: 10/31/2016)

From James Larson:

Team,

Thank you for your work on lightweight crypto.
In the template copied and pasted below, I suggest that you insert the line, Design Goals, after the line Primitive.
The reason is that all of the characteristics that follow in the template; Physical, Performance and Security, will be driven by the goals.

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	*Type of primitive*
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.*
Design goals	*List design goals.*

531

Thank you, again, for your work.
I hope that my suggestion helps the team.

Jim

James Larson
Management Information Specialist
U.S. Department of Housing and Urban Development
Housing Office of Systems and Technology

From Thomas Peyrin:

Hello,

I believe there is an important criterion that need to be taken in account and that doesn't appear in your document. In many cases, the constrained devices will discuss with a server. This server will potentially have thousands of devices communicating with it simultaneously. Therefore, it is important that the algorithm performs well in software (and I don't mean microcontrollers, but high-end software platforms), so that the server doesn't get quickly overloaded.

Moreover, the problem has to be checked in details, as it is probable that all devices will send very small amount of information at a time, all devices having a different key. Therefore, common bitsliced implementations will surely not perform as good as they are supposed to. This is what we have started to investigate in this paper: https://eprint.iacr.org/2013/445

As a side note, I also believe that having both encryption and decryption at the same time is not always needed. Actually, if you are very constrained on your device, you will probably end up implementing only the exact function that the device has to perform (encryption-only or decryption-only or MAC-only) to save as much area as possible.

Regards,

Thomas.

From Hirotaka Yoshida:

-----1. Introduction

1)

It would be helpful for the readers if you explain relevant application/systems information more specially and clearly. This can be done by including: "The portfolio of NIST lightweight cryptographic primitives could be considered to be used as the cipher suite in the NIST standards such as NISTIR 7628 Guidelines for Smart Grid Cyber Security http://www.nist.gov/smartgrid/upload/nistir-7628_total.pdf"

Other important application is automotive but I do not know if NIST published something on automotive security.

2)

I understand that lightweight public-key (LW PK) and lightweight protocol is out of scope and that the initial scope is symmetric crypto. I think this approach is reasonable because LW symmetric looks more mature to me. But I still think that it would be better if you include a big picture or roadmap of the NIST lightweight crypto project. For instance, you may want to say something like: "After this first project that may end in

20XX and NIST may consider the similar project on protocol, finally and LW PK."

The back ground behind this is there is rapidly increasing demand from automotive industry. I understand that V2V(vehicle-to-vehicle) communication needs security using PK. It would be wise not to ignore this trend.

-----2.5 Lightweight cryptography standards
3)
You might be interested in RFID standard including cipher suites,
PRESENT-80 and Grain-128A specified in ISO/IEC 29167-10 -- 29167-19.
http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=45332&published=on

4)
The reader would confirm that 29192-6 project exists by
referring to:
http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=71116
****

Best regards,
Hiro

From Stephen Farrell:

Hi,

I have the following comments:-

- (A minor point) I think AES-equivalent security is by far the more interesting measure and not 112 bit security. Perhaps aims-for-128-bit-security would be ok.

- One could generalise from that and state that since almost all devices and absolutely all widely deployed crypto algorithms will "touch" the Internet, that the real security requirement for crypto is "no worse crypto than used in TLS in a typical browser/web-server." If anything worse is chosen then one is introducing a major weak-link.

- Further to the above, the document should note that any crypto ever used to talk to a device that connects to the Internet is really subject to the Internet threat model and hence cannot use weaker crypto than would be acceptable on the Internet.

- Almost all devices using a supposed "lightweight" crypto solution will run s/w and will have to have a way to update their s/w and hence will require something like a digital

signature or data origin authentication that is cryptographically strong enough to resist the most interesting attack that can be conceived against any device using our putative lightweight algorithm. Given that code-footprint is a significant constraint in many devices that might choose a lightweight algorithm, it seems the one can conclude that there is little to no reason to ever want to weaken the data integrity or data origin authentication requirements. I think that also means that there is no merit in arguments that one can do without the code for "strong" crypto. I think a corollary is that there is less scope for real uses of supposed lightweight crypto than appears to be the case if one ignores software update. And one ignores software update at all our peril. The document should note that the existence of a need for strongly-protected s/w update on the same device ought be a base assumption in design of any supposed lightweight crypto algorithm.

Regards,
Stephen Farrell.
https://www.cs.tcd.ie/Stephen.Farrell/

From Mark D. Aagaard, Guang Gong, Kalikinkar Mandal:

We strongly support NIST's Information Technology Laboratory's lightweight cryptography project and the effort to work toward standardization of lightweight cryptographic algorithms. The 2015 workshop was valuable in providing a venue for presenting research focused on lightweight cryptography and a gathering to facilitate discussions between researchers and practitioners. We are looking forward to 2016 workshop.

In the following, we refer to *Draft NISTIR 8114: Report on Lightweight Cryptography* as "the report".

First, we should say that we agree with almost everything in the report. The overall process to move toward standardization, target devices, the use of profiles, the metrics, *etc* are all carefully thought out and well presented. Our comments below focus on areas where we think the report could be improved.

1 Top-Level Comments

Comment 1: The report mentions the standards for EPC tags, but it would be helpful to mention other related standards, such as:

- NFC tags: ISO-14443
- Zigbee
- Vehicles:
    - IEEE Standard 1609.2: Wireless Access in Vehicular Environments (WAVE), Security Services for Applications and Management Messages
    - IEEE Draft P802.11-REVmb/D11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification
- Medical: this is just an emerging area, but a potential standard of some relevance is IEC 60601-2- 24:2012-Ed.2.0

Comment 2: The report should list the security functions that are relevant to lightweight cryptography and should list security functions that are not relevant. A list of functions to considered could include:

- entity authentication (privacy preserving or not)
- en/de-cryption
- message authentication
- authenticated encryption
- attestation
- digital signatures

Comment 3: The report should provide a minimum expected lifespan of usefulness for a primitive. For example, with the rapid pace of changes in this area, should a primitive proposed today be required to still be relevant 10, 20, or 50 years from now?

Comment 4: The report should provide a maximum expected time until deployment for a primitive. For example, should an primitive be implementable with today's technology, or is it acceptable for a primitive to be dependent upon technological advances that might not be achievable for 5 or 10 years?

Comment 5: 112-bit security might be overkill for some applications, such as low-value EPC tags.

Comment 6: The report should include stream ciphers in the initial focus of the project on lightweight cryptography.

Rationale: Lightweight stream ciphers Grain v1, Trivium and MICKEY 2.0 are the finalists of the ECRYPT Stream Cipher Project (in 2008). Only Grain v1 and MICKEY

2.0 provides 128-bit security, and Trivium provides 80-bit security. As NIST targets to provide at least 112-bit security, there is a need for new stream ciphers for lightweight applications. With suitable modes of operations, message authentication codes (MACs) and pseudorandom number generators (PRNGs) can be constructed from stream ciphers with small amount of hardware resources.

Comment 7: The report should include Pseudorandom Number Generator as a primitive.

Rationale: The EPC Gen2 standard uses random number generators for generating 16-bit random numbers in the inventory and tag access protocol. As RFID is a rapidly evolving field, the length of messages to be transmitted are increasing and the security functionalities are being added over time. For instance, in Version 1 of the EPC Gen2 standard, only PRNGs were needed, but in Version 2 recommends a PRNG and an encryption algorithm.

2 Physical Metrics

Comment 8: The report should measure throughput in bytes per clock-cycle, latency in clock cycles. Measurements of power should be replaced by energy-per-byte. Primitives should specify their range of typical clock speeds.

Rationale: The maximum clock speed of a lightweight primitive is usually limited by constraints on the maximum power consumption, not the delay of the circuitry. The clock speed is not an intrinsic property of the cryptographic primitive, but instead is a parameter than can be adjusted to make tradeoffs between performance and power consumption. The intrinsic properties of a primitive are the throughput and latency as measured in clock cycles and the energy consumed per operation.

Comment 9: The report should measure both the average throughput of a primitive and "burstiness" of the throughput.

Rationale: For example, a block cipher may have an average throughput of 1 byte / 8 clock cycles, but output nothing for 63 clock cycles and then output a full 8-byte block in a single clock cycle. For many systems that wish to use lightweight cryptography, it will be simpler to use a primitive that has a constant throughput of 1 byte / 8 clock cycles. A bursty primitive will incur an additional cost in hardware for system integration. The current draft report already hints at this in question 11 (line 502), which asks "is the platform an inherently serial one, or can data be processed in parallel?"

Comment 10: The report should use a neutral measure of area for FPGAs, such as a

generic FPGA cell with a flip-flop and either 6-input or 4-input LUT.

Rationale: “Slice” is a term used specifically by Xilinx. Another leading FPGA vendor, Altera, uses “adaptive logic module” (ALM). Both slices and ALMs contain multiple lookup tables and flip-flops. As the report notes, the specifics of what is contained in a slice/ALM differs from chip-to-chip, making it difficult to compare results. Most commercial synthesis tools have the ability to report area in terms of a generic FPGA cell.

Comment 11: The report should suggest a technology node or feature size (*e.g.*, 65 nm, 90 nm, or 130 nm) for ASIC implementations.

Rationale: A consistent feature size will make it easier to compare results. The feature size should be chosen so that results will be relevant to industry and be accessible to academic researchers. Most academics today will have access to 130 nm cell libraries and many will have access to 90 nm and 65 nm. Access to smaller feature sizes is usually more difficult for academics, because of the proprietary nature of the cell libraries.

Comment 12: The report should propose one or more Figures of Merit that could be used in com- paring primitives across multiple metrics. Typical figures of merit are: throughput/area, throughput/area^2, throughput/(area×power). Alternatively, a profile could include a suggested figure of merit.

3 Profiles and Primitives

Comment 13: A profile should specify both the expected maximum number of messages per key and the expected maximum length of a single message.

Comment 14: The report should specify a required or recommended set of block cipher modes.

Rationale: It might be possible to optimize a block cipher to support only one or a few different modes. Lightweight cryptographic systems tend to be used in more application-specific ways that general crypto- graphic systems, and so it may be feasible and advantageous to incorporate the mode circuitry/software into the primitive itself.

Comment 15: The report should specify a required or recommended set of operations that each primitive supports. These operations could be based on FIPS-140 (*e.g.*, load-key, save-key, erase-key, *etc*).

Comment 16: The report should specify requirements or recommendations for when input or output data must be transmitted contiguously or may be provided intermittently. When data may be transmitted intermittently, the report should specify what granularity must be contiguous (*e.g.*, byte, word, or block).

Rationale: It is usually more complicated or expensive for a primitive to support intermittent data trans- mission, but in the real world, it may not be possible to provide a contiguous stream of data. Even in loading the key, the system may require multiple clock cycles to fetch each byte or word of the key from memory and send it to the primitive. In hardware, intermittent data transmission requires registers with chip-enables or clock-gating circuitry, which can consume extra area in an ASIC.

Comment 17: It would be beneficial if the report could specify a recommended standard interface for each category of primitive.

From Hongjun Wu:

Dear NIST,

In the report, it is stated that "the initial focus of the project is on block ciphers, hash functions, and message authentication codes".

My comment: I think that NIST may consider the lightweight authenticated-encryption primitives and modes in this project. When both encryption and authentication are needed (since encryption always requires authentication), the authenticated encryption algorithm can be more lightweight and more efficient than implementing encryption and authentication separately.

Best regards,
hongjun

From NSA:

Here are our comments on the draft of NISTIR 8114, dated August 2016. Substantive comments on the contents are upfront, followed by minor (editorial) comments.

Substantive Comments

Lines 185-190: Low latency is somewhat overemphasized in this document. There are likely niche applications where hardware latencies in the 15-20 ns range are desirable. But for the IoT applications noted, these figures are orders of magnitude smaller than necessary. For example, profile 1, exemplified by the RFID asset tracking application, asserts that a 15 ns latency is required. This figure does not appear to be supportable; we would appreciate hearing from the RFID community what the actual requirement is. Profile 2 requires a 20 ns latency. The application cited is command validation on a Controller Area Network (CAN) bus. This figure is also hard to justify. First, in automotive applications the CAN bus tends to be microcontroller based, operating at around 500 kbps, and therefore cannot achieve anything remotely close to a 20 ns latency. For automotive applications, time scales tend to be in the milliseconds rather than nanoseconds (a factor of about 10^6): The decision to deploy an airbag in a crash is made within 15-30 milliseconds after the onset of the crash (see wikipedia.org/wiki/Airbag). The deployment takes between 20-30 milliseconds. Why does authentication need to happen at one-millionth this amount of time? Would a latency of, say, 1 microsecond not suffice in this case? A study performed at the University of Maryland Automotive Testing Laboratory attempted to determine what an adequate latency for a block cipher would be for CAN bus applications. A report will be provided to NIST. The report concludes that 500 microseconds, 25000 times slower than the number provided in sample profile 2, is adequate and that software implementations of a block cipher may suffice. Many similar examples could be given; we have not seen applications on constrained platforms where the latency figures cited are necessary or beneficial. For all of the applications we have investigated, block cipher latency is a nonissue.

Lines 264-267: While generating keys with a KDF is generally considered good security practice, it's not clear why generating keys with a KDF (or any other secure method of generating keys pseudo-randomly) would necessarily prevent an attack based on weak keys, which is one of the cases listed. While a KDF will prevent one from explicitly choosing weak keys, weak keys will occur at random according to the size of the weak

key class(es). Perhaps it is being assumed here that the size of the weak key space is small, so that generating weak keys with a KDF would be unlikely. However, the size of the weak key space could be reasonably large, in which case generating keys with a KDF will not prevent weak keys from occurring at random, provided enough keys are generated. It may be appropriate to mention or discuss this briefly; otherwise, a reader may get the false impression that KDFs can completely prevent/block any attack based on weak keys.

Lines 337-342: Unlike the discussions in the previous paragraphs of section 2.4 on block ciphers and hashing, there is no mention of the performance of authenticated encryption and MACs in resource-constrained environments. Given the title of section 2.4, perhaps it could be noted that the performance of these modes in resource-constrained environments depends on the performance of block ciphers and hash functions (which was discussed in the previous paragraphs), since they are typically used as the underlying cryptographic primitives in these modes.

Line 347: It might be mentioned that SIMON and SPECK are currently being included in an amendment to ISO/IEC 29192-2, and that LEA has been proposed as an amendment to part 2 of ISO/IEC 29192-2.

Line 390: We believe that the primary characteristics of lightweight cryptography are size of implementation (in hardware or software) and power/energy consumption, neither of which is listed in the criteria. We also believe throughput is an important design criterion as most lightweight applications will encrypt multiple blocks and performance is desirable even with lightweight devices. It should be clearly stated that

“Algorithms and modes that do not have related-key attacks are preferred” and “Algorithms and modes that do not have limits on the plaintext/ ciphertext pairs are preferred”, as these do not weaken the security of the algorithm.

Lines 449-452: Given that 112 bits of security against key recovery attacks is being emphasized here, does this mean that any key recovery attack must require at least 112 bits of work, even if the attack requires more plaintext/ciphertext pairs than might be allowed by the profile? For example, suppose an algorithm has a limit of 2^{48} plaintext/ciphertext pairs under a given key. If a key recovery attack requiring, say, 2^{96} work and 2^{52} plaintext/ciphertext pairs existed, would the algorithm still be viewed as providing 112 bits of security against key recovery attacks? Since a limit on the number

of plaintext/ciphertext pairs is listed as one of the possible desired properties that NIST will use to evaluate designs, it may be helpful to discuss how the overall security against key recovery will be measured if an attack exceeds those limits. Probability of success of the attack may need to be taken into account as well. In the example above, it could be the case that if the limit of 2^{48} plaintext/ciphertext pairs is respected, then the attack succeeds with probability 1/16. Again, with a work factor of 2^{96}, would the algorithm still be viewed as providing 112 bits of security against key recovery attacks?

Lines 449-452: Even-Mansour block ciphers, such as PRINCE and Chaskey, have a significantly weakened security model where if a single key encrypts any D known plaintexts the work required to perform a key recovery attack is reduced by a factor of D. Line 451 states security against key recovery attacks should be at least 112 bits, which means if we have an Even- Mansour design with a 128-bit key, we must enforce a data limit of 2^{16} blocks.

The security of an Even-Mansour design is not the same as its key size, which is different to classical block ciphers. Engineers, who are not typically cryptographic experts, may mistakenly assume a 128-bit Even-Mansour block cipher is equivalent to AES128 as it has the same block and key size, when it is significantly weaker.

The security proofs [Mouha, Luykx] for Even-Mansour designs in the multi-key setting show that the work to attack one of a number of keys remains linear in the combined data they produce. Data can be collected across multiple users to increase the attacker's advantage. Mavromati showed that 2^{32} users each encrypting 2^{32} blocks of data gives an attack with $O(2^{64})$ work that recovers a pair of keys. Biham's generic attack on AES would still take $O(2^{96})$work in this case.

Line 453: Profiles should not be written in order to include or exclude particular algorithms. Rather they should be written with careful consideration given to the needs of the application. Care should be taken to develop realistic profiles with realistic performance characteristics. When citing example usages, every effort should be made to ensure that the profile characteristics are actually appropriate for the example, not too loose and not overly restrictive. This will require some research and there should be some justification for the characteristics.

Profiles should be written to cope with changes in technology. Specifying performance requirements based on current industry applications will lead to a standard that rapidly

becomes outdated and unfit for purpose.

Profile Sample_3: One might contend that “resistance against tag forgeries” (which is listed as a design goal) is inherent for any MAC and thus, does not need to be listed explicitly. Furthermore, it is not clear why this is listed as a design goal. It would seem that such resistance is a security characteristic, rather than a design goal. Note that in both of the first two sample profiles, resistance against certain types of attacks is listed as a security characteristic, rather than a design goal.

Editorial Comments

Line 65: “...the findings of ~~the~~ NIST’s lightweight cryptography...”

Line 116: “...to ~~get~~ solicit public feedback on the constraints...”

Line 121: “...cryptography project and to outline...”

Line 144: Place a colon after “Figure 1” in the caption.

Line 156: “...with 64 bytes of RAM ~~and~~ or less, going down...”

Line 168: “...exhaustive list, but instead to illustrate...”

Line 168: Insert “an” between “be” and “exhaustive”.

Line 187: “...of time between the initial request of an operation...”

Line 189: “...the initial request for the encryption of a plaintext...”

Line 198: Make “implementation” plural.

Line 204: Make “GE” plural.

Line 220: Replace “crypto” with “cryptographic”.

Line 221: “...have been proposed ~~to bring~~ which offer performance advantages...”

Line 226: “...are weak – rather, the idea is to use...”

Line 227: “...advancements ~~to~~ that result in designs...”

Line 239: Insert “the” between “from” and “1990s”.

Line 246: Replace “length of the plaintexts” with “number of plaintext blocks”.

Line 257: Insert a space between “28” and “GEs”.

Line 258: Insert “the” between “whereas” and “AES”.

Line 276: “...smaller internal states and output sizes might be used.”

Lines 302-303: It is not clear what is meant by the phrase “...and its security mostly depends on the hardness of analysis.”

Line 303: “...flexibility and is susceptible to timing...”

Line 308: “block cipher algorithms; AES and Triple DES...” Replace the semicolon with a colon.

Line 311: “...AES variants ~~operate~~ have a block size...”

Line 315: “...has been achieved in 124.6 cycles per byte ~~124.6~~ and decryption...”

Line 324: “...specifies SHA-1, the SHA-2 family...” Remove the comma and replace with “and”.

Line 325: “...SHA-512-224 and...” Replace with “SHA-512/224”.

Line 342: “...generating and verifying tags to provide message authentication.”

Line 364: Insert “the” between “on” and “state”.

Line 349: “...asymmetric techniques, namely...”

Line 365: “...lightweight cryptography and its applications, ~~and~~ performs implementation...”

Line 393: “...block ciphers, authenticated encryption schemes, hash functions, and...”

Lines 398-399: “....quantum attacks, ~~or~~ and 2) they use a combination of general...”

Line 399-400: Insert “a” between “e.g.,” and “lightweight”.

Line 422: Replace “amount of data” with “amount of output data (compared to the amount of input data)”.

Line 426-427: Insert “as” between “such” and “timing”.

Line 434: Insert “processed” after “pairs”.

Line 435: “...constraints of the devices, (e.g., limitations...” Remove the comma immediately after “devices”.

Line 436: “...processed by the same key), or by message formats defined...”

Line 442: “...where keys are not chosen independently and...”

Line 448: “...attacks that require a large number...”

Line 528: Insert a forward slash (/) between “characteristics” and “goals”.

From the Ketje and Keyak Team:

Dear NIST,

here are the comments of the Keccak team to "Report on Lightweight Cryptography", draft NISTIR 8114 of August 2016.

1) Focus on functionality rather than primitives NIST should focus on functionality (e.g., encryption, authentication, authenticated encryption, hashing, key derivation) rather than on primitives. Functionality is almost always realized by applying modes to primitives.

1.a) Primitives should include permutations.

1.b) Modes shall be considered separately. The cost of the mode has an important impact on the total cost. E.g., a stand-alone block cipher is of no use and requires a mode of operation. When comparing a solution based on a block cipher and a solution based on a permutation using the sponge and duplex constructions, one can see that in the latter case the mode introduces less overhead. See, e.g., the presentation of Jens-Peter Kaps at DIAC 2014: http://2014.diac.cr.yp.to/slides/kaps-keccak.pdf

2) Multi-target security strength

When specifying security strength (Section 3.1.1), it should be made clear whether multi-target security is meant. We think requiring security strength in multi-target scenarios is appropriate.

3) Take into account protection against side channel attacks

Protection against side channel attacks is crucial in many use cases that require lightweight cryptography. It should be taken into account by defining metrics (power, energy, area, RAM consumption, etc.) of implementations that offer a certain level of resistance against side channel attacks such as power analysis, electromagnetic analysis, timing attack and fault attacks. This can be made explicit in Section 2.2 and in the Table on page 17.

4) Energy per bit or per message as metric in the table on page 17, energy is missing. Relevant is energy per message bit and minimum energy consumed for treating a message.

5) Resistance against related-key attacks

Resistance against related-key attacks has a cost and no added value when the keys are randomly drawn or derived from a decent KDF. About related-key attacks, the text states on lines 445-447 (p 10-11) "This attack model still remains highly relevant for devices where the capability to update the key exists." This is only true if a key is updated by a related key and independent of the capability to update the key.

Kind regards,

Gilles, Guido, Joan, Michaël and Ronny

From CDC:

CDC has no comments to provide on the *Draft NISTIR 8114, Report on Lightweight Cryptography*.

Thank you for the opportunity to review and comment.

Michael Harris, CISSP Information Technology Specialist (Information Security)
Centers for Disease Control and Prevention