NIST's Lightweight Crypto Standardization

Meltem Sönmez Turan

Flexible Symmetric Cryptography, Leiden, The Netherlands

March 19, 2018

Outline

- NIST's Computer Security Division
- Update on the Lightweight Cryptography Project
- Some open research problems

NIST Computer Security Division

National Institute of Standards and Technology

- Non-regulatory federal agency within U.S. Department of Commerce.
- Founded in 1901, known as the National Bureau of Standards (NBS) prior to 1988.
- Headquarters in Gaithersburg, Maryland, and laboratories in Boulder, Colorado.
- Employs around 3,000 employees, and 2700 associates.

National Institute of Standards and Technology

- Non-regulatory federal agency within U.S. Department of Commerce.
- Founded in 1901, known as the National Bureau of Standards (NBS) prior to 1988.
- Headquarters in Gaithersburg, Maryland, and laboratories in Boulder, Colorado.
- Employs around 3,000 employees, and 2700 associates.

NIST's Mission

to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life.

NIST Organization Chart

Seven Laboratory Programs

- Center for Nanoscale Science and Technology
- Communications Technology Lab.
- Engineering Lab.
- **Information Technology Lab.**
- Material Measurement Lab.
- NIST Center for Neutron Research
- Physical Measurement Lab.

Information Technology Lab.

- Advanced Network Technologies
- Applied and Computational Mathematics
- Applied Cybersecurity
- **Computer Security**
- Information Access
- Software and Systems
- Statistical Engineering

Computer Security Division

- Cryptographic Technology
- Secure Systems and Applications
- Security Outreach and Integration
- Security Components and Mechanisms
- Security Test, Validation and Measurements

Computer Security Division (CSD)

Conducts research, development and outreach necessary to provide standards and guidelines, mechanisms, tools, metrics and practices to protect nations information and information systems.

Who does CSD work with?

- Government: Core user community.
- Industry: On adoption of cryptographic algorithms, feedback mechanism on standards.
- Academic Researchers: Development of new algorithms/modes schemes to advance science of cryptography.
- Standards Developing Organizations: Adoption and development of new standards.

CSD Publications

- Federal Information Processing Standards (FIPS): Specify approved crypto standards
- NIST Special Publications (SPs): Guidelines, technical specifications, recommendations and reference materials, comprising multiple sub-series:
  - SP 800 - Computer security
  - SP 1800 - Cybersecurity practice guides
  - SP 500 - Information technology (relevant documents)
- NIST Internal or Interagency Reports (NISTIR):Reports of research findings, including background information for FIPS and SPs
- NIST Information Technology Laboratory (ITL) Bulletins: Monthly overviews of NIST's security and privacy publications, programs and projects

to subscribe for publication announcements:

`https://public.govdelivery.com/accounts/USNIST/subscriber/new?qsp=USNIST_3`

How does CSD develop standards?

- International "competitions": Engage community through an open competition (e.g., AES, SHA-3, PQC).
- Adoption of existing standards: Collaboration with accredited standards organizations (e.g., RSA, HMAC).
- Open call for proposals: Ongoing open invitation (e.g., modes of operations).
- Development of new algorithms: if no suitable standard exists (e.g., DRBGs).

Update on the Lightweight Cryptography Project

NIST's Lightweight Crypto Project

Motivation

- Shift from general-purpose computers to dedicated resource-constrained devices.
- New applications that use private information
- Lack of crypto standards that are suitable for constrained devices.

Goal

- Understanding the need for lightweight crypto.
- Developing new lightweight crypto standards.

What is Lightweight Cryptography?

- Subfield of cryptography that aims to provide crypto solutions tailored to constrained environments.
- Lightweight cryptography does not mean weak crypto.
- Security properties may be different than those desired for general use, but must be sufficient for the target application.
- Lightweight crypto may
  - be less robust,
  - be less misuse resistant,
  - have fewer features.

So far ...

- The project started in 2014.
- First Lightweight Crypto Workshop at NIST, July 2015.
- Second Lightweight Crypto Workshop at NIST, October 2016.
  - Consensus that there is need for lightweight crypto standards.
- Published *NISTIR 8114 - Report on Lightweight Cryptography,* March 2017 (after 90-day comment period).
- Published *(draft) Profiles for the Lightweight Cryptography Standardization Process*, April 2017.

NISTIR 8114 - Report on Lightweight Cryptography

Report provides information on:

- Overview of lightweight cryptography: Target devices, performance metrics, lightweight primitives, performance of NIST standards in constrained environments, and other lightweight crypto standards.
- NIST's Lightweight Crypto Project: Scope, design considerations, profiles, and evaluation process.

NIST's Lightweight Crypto Project

Scope:

- All cryptographic primitives and modes that are needed in constrained environments.
- Initial Focus: Symmetric Cryptography.

Standardization Plan:

- Develop and maintain a portfolio of lightweight algorithms and modes that are approved for limited use.
- Use *profiles* to specify algorithm requirements.
- The lightweight portfolio is not intended to offer alternative algorithms for general use. Conventional crypto standards should be used when their performance is acceptable.

Summary of the Comments Received on NIST 8114

- Including stream ciphers, authenticated encryption schemes, and permutations to initial focus (block ciphers, hash functions, MACs).
- Using 'functionality' instead of 'primitives'.
- Emphasis on the importance of performance on high-end SW platforms as the constrained devices will communicate with a server.
- Emphasis on the importance of efficiency of software updates (the need for digital signatures and hashing).
- Comments on the size of the keys.
- Emphasis on the importance of side channel attacks.

Profile Template

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)
Security characteristics	Minimum security strength, relevant attack models, side channel resistance requirements, etc.

Developing Profiles

NISTIR 8114 included 22 questions to industry partners to understand their lightweight crypto need.

- Application: Target functionality? Typical plaintext, tag sizes?
- Constraints: What limits are imposed on the energy and/or power that is available to the device? Does the device have to respond within a specific time?
- Cryptographic keys: How are keys generated? Where are they stored, and for how long? How much data is processed under the same key?
- Software implementations: Which platforms? Which specific types of processors? Limits on the amount of registers, RAM and ROM? Is it necessary to obfuscate the implementation?
- Hardware implementations: Which types of hardware are considered (FPGA, ASIC, etc.)? Limits on the amount of logic blocks or GEs?
- Side channel resistance: Side-channel or fault attacks required?

Responses to the Questionnaire

- Target applications: Hardware encrypted data storage device, low-cost and low-consumption sensor data transmission, RAIN RFID tags for anti-counterfeiting solutions, IoTs, wearables, low power wireless sensor networks.
- Target functionality: Encryption, AE, hashing, key agreement, sensor/tag authentication, with plaintext size of 16 bytes
- Target devices: ARM Cortex-M0 processors, Intel Quark SoC X1021, Atom E3826
- Side channel resistance: In general, good to have.

Developing Profiles

Due to the variability of the requirements, use cases, and target devices, having a single profile may not be optimal.

Biryukov and Perrin [1] proposed to split into two areas

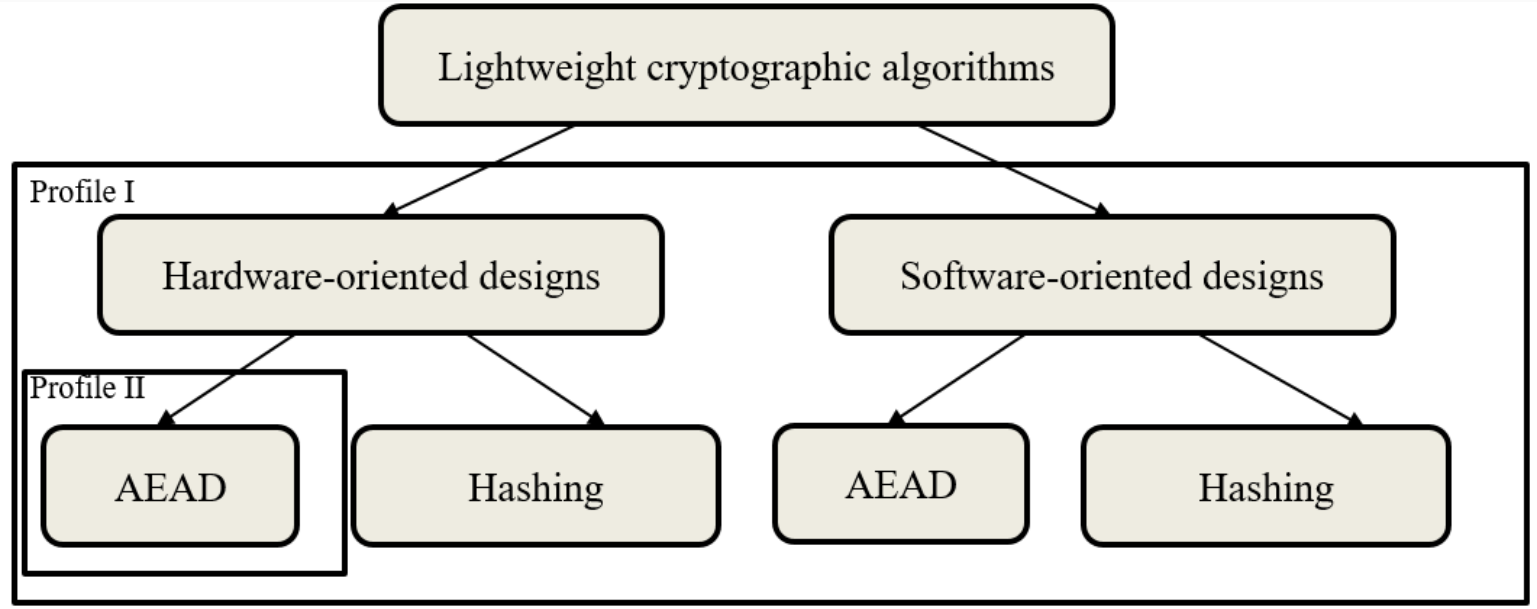
- Ultra-lightweight cryptography: lower security level, for more constrained devices,
- IoT/Ubiquitous cryptography: similar to the requirements of conventional crypto.

[1] A. Biryukov, L. Perrin, *State of the Art in Lightweight Symmetric Cryptography*, IACR Cryptology ePrint Archive, 2017:511, 2017

Draft Profiles for Lightweight Cryptography

NIST published Draft White Paper *Profiles for Lightweight Cryptography Standardization Process* in April 2017.

- Profile I Authenticated Encryption with Associated Data (AEAD) and Hashing for constrained software and hardware environments.
- Profile II AEAD for constrained hardware environments.



Profile I - AEAD and Hashing for Constrained Environments (1/4)

Functionality:

- Authenticated encryption with associated data and hashing

Design goals:

- Performs significantly better in constrained environments (hardware and embedded software platforms) compared to current NIST standards.
- Both algorithms should be optimized to be efficient for short messages (e.g., as short as 8 bytes).
- The message length shall be an integer number of bytes.

Physical characteristics

- Compact hardware implementations and embedded software implementations with low RAM and ROM usage should be possible.

Profile I - AEAD and Hashing for Constrained Environments (2/4)

Performance characteristics:

- The performance on ASIC and FPGA should consider various standard cell libraries, the flexibility to support various implementation strategies (low energy, low power, low latency), with significant improvements over current NIST standards.
- The performance on microcontrollers should consider a wide range of 8-bit, 16-bit and 32-bit microcontroller architectures.
- The preprocessing of a key (in terms of computation time and memory footprint) should be efficient.

Profile I - AEAD and Hashing for Constrained Environments (3/4)

Security characteristics for AEAD

- A key length of 128 bits shall be supported.
- Nonce and tag lengths of up to 128 bits shall be supported.
- Plaintext and associated data lengths of up to $2^{50} - 1$ bytes shall be supported.
- At least $2^{50} - 1$ bytes can be processed securely under a single key.
- Cryptanalytic attacks should require at least 2^{112} computations on a classical computer in a single-key setting.
- Lends itself to countermeasures against various side-channel attacks, including timing attacks, simple and differential power analysis (SPA/DPA), and simple and differential electromagnetic analysis (SEMA/DEMA).

Profile I - AEAD and Hashing for Constrained Environments (4/4)

Security characteristics for hashing

- Cryptanalytic attacks should require at least 2^{112} computations on a classical computer.
- Hash outputs of 256 bits must be supported, and longer hash values may be supported as well.
- A maximum message length of $2^{50} - 1$ bytes shall be supported.
- Lends itself to countermeasures against various side-channel attacks, including timing attacks, simple and differential power analysis (SPA/DPA), and simple and differential electromagnetic analysis (SEMA/DEMA).

Profile II- AEAD for Constrained Hardware Environments (1/2)

Functionality:

- Authenticated encryption with associated data

Design goals:

- Performs significantly better compared to current NIST standards.
- The performance for short messages (e.g., as short as 8 bytes) is important.
- The message length shall be an integer number of bytes.

Physical characteristics

- Targeted towards constrained hardware platforms.
- Compact hardware implementations should be possible.

Profile II - AEAD for Constrained Hardware Environments (2/2)

Performance characteristics:

- The performance on ASIC and FPGA should consider a wide range of standard cell libraries and vendors.
- Algorithm should be flexible to support various implementation strategies (low energy, low power, low latency).
- The preprocessing of a key (in terms of computation time and memory footprint) should be efficient.

Security characteristics:

- Same as security characteristics of AEAD in Profile I.

Current Status

- Developing the submission, evaluation, and selection processes (e.g., submission requirements, API, evaluation methods).
- Preparing a single call for submissions for both profiles: AEAD with optional hashing.

Draft Submission Requirements and Evaluation Criteria

- Requirements to be a complete submission
  - Cover sheet, specification, supporting documents, source code, test vectors, and IP statements
- Requirements to be a proper submission
  - AEAD security requirements
  - Hash function security requirements
  - Design requirements
  - Additional AEAD+hashing requirements
  - Implementation requirements

AEAD Security Requirements (1/2)

An authenticated encryption with associated data (AEAD) algorithm is a function with four byte-string inputs and one byte-string output. The four inputs are a variable-length plaintext, variable-length associated data, a fixed-length nonce, and a fixed-length key. The output is a variable-length ciphertext.

- Confidentiality of the plaintexts (under adaptive chosen-plaintext attacks) + Integrity of the ciphertexts (under adaptive forgery attempts)
- Nonce is assumed to be unique under the same key.
- Similarities with CAESAR call for submission

AEAD Security Requirements (2/2)

- Family of algorithms
  - One primary member with key, nonce and tag lengths of 128, 96 and 64 bits, respectively.
  - Limits on the input sizes for the primary member **shall not** be smaller than $2^{50} - 1$.
  - Family can include at most 10 members.
- Keys sizes **shall** at least be 128 bits. Attacks **shall** require at least 2^{112} computations. If larger key sizes are supported, it is recommended that at least one member has key size of 256 bits (and resistance to attacks **shall** at least be 2^{224} computations).
- Well-understood, and analyzed. Submissions are expected to have third-party analysis.

Hash Function Requirements

A hash function is a function with one byte-string input and one byte-string output. The input is a variable-length message. The output is a fixed-length hash value.

- Computationally infeasible to find a collision or a (second) preimage. Resistance to length extension attacks.
- Cryptanalytic attacks on the hash function **shall** require at least 2^{112} computations on a classical computer.
- The hash function **shall not** specify hash values that are smaller than 256 bits.
- Family of algorithms
  - One primary member has a hash size of 256 bits.
  - Limits on the input sizes for the primary member **shall not** be smaller than $2^{50} - 1$.
  - Family can include at most 10 members.

Additional Requirements for Submissions with AEAD and Hashing

- Submissions **shall** state which design components the AEAD and hashing algorithms have in common, and explain how these common components lead to a reduced implementation cost.
- Submissions **shall** specify list of pairs of AEAD and hash function family members to be evaluated jointly. This list is permitted to be as short as one recommendation. Primary member of the AEAD family and primary member of the hash function family **shall** be paired together. This list **shall not** be longer than ten recommendations.

Design Requirements

- Submissions **shall** perform significantly better in constrained environments (HW and SW platforms) compared to NIST standards.
- Optimized to be efficient for short messages.
- Implementations should lend themselves to countermeasures against various side channel attacks.
- Designs can make tradeoffs between performance metrics, and submitters are allowed to prioritize certain performance requirements over others.

Implementation Requirements

- Reference software implementation in C, to support public understanding.
- An implementation **shall** be provided for all variants.
- Code **shall not** contain compiler intrinsics, platform-specific headers, or compiler-specific features.
- API is compatible with eBACS: ECRYPT Benchmarking of Cryptographic Systems.
- Submission may include optimized implementations that use the same API, or additional implementations that highlight specific implementation features of the algorithms. There are no restrictions on the API for the additional implementations.
- The correctness of the reference implementation will be verified on the NIST test vector verification platform.

Evaluation Process and Tentative Timeline

- Submissions will be analyzes based on security, performance and also side channel resistance.
- Submissions that have significant third-party analysis or leverage components of existing standards will be favored for selection.

Tentative timeline

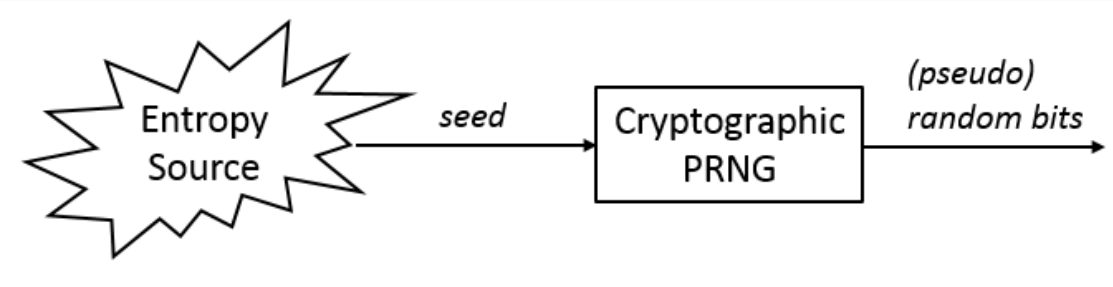
- March 2018, publish draft call for submission.
- June 2018, publish final call for submision.
- December 2018, deadline for submission (tight deadline).
- NIST will publish the complete and proper submissions.
- Initial evaluation 12 months.
- Workshop will be held ten to twelve months after the submission deadline.
- Standardization within two to four years, after the public analysis starts.

Some open research questions

Open Research Related to SP 800 90 Series

NIST SP 800 90 Series - Recommendations on Random Number Generation

- 90A - Deterministic RNGs
- 90B - Entropy Sources
- 90C - RNG Constructions



Entropy Estimation - 90B Perspective

- 90B aims to estimate entropy of noise/entropy source outputs.
- Black box analysis, based on different statistical assumptions, and minimum estimate is awarded.
  - Black box analysis is necessary for lab evaluation. Although it is not the optimal strategy, it improves the quality of the RNGs.
- We propsoed the predictors framework to estimate entropy [2].

Open issues include

- Designing noise source specific predictors, Simulating and modeling TRNG outputs, using multiple noise sources.

[2] Kelsey J., McKay K.A. and Sonmez Turan, M., Predictive Models for Min-Entropy Estimation, Workshop on Cryptographic Hardware and Embedded Systems, CHES 2015, Saint Malo, France, 2015

Circuit Complexity

Given a set of gate types, construct a circuit that computes f and is optimal to some criteria

Multiplicative Complexity is the minimum number of AND gates that are sufficient to evaluate the function over the basis (AND, XOR, NOT).

Why do we count the AND gates?

- Lightweight Cryptography: The technique of *minimizing the number of AND gates, and then optimizing the linear components* leads to the implementations with low gate complexity.
- Secure multi-party computation: Reducing the number of AND gates improves the efficiency of SMP protocols.
- Side channel attacks: Minimizing the number of AND gates is necessary when implementing a masking scheme to prevent side-channel attacks.

Multiplicative Complexity is affine invariant!

Affine Equivalance

An affine transformation from g to f in B_n is a mapping of the form

$$f(x) = g(Ax + a) + b \cdot x + c,$$

where A is a non-singular $n \times n$ matrix over $\mathbb{F}_2$; x, a are column vectors over $\mathbb{F}_2$; b is a row vector over $\mathbb{F}_2$.

- f, g are affine equivalent, if and only if there exists an affine transformation between them.
- Affine equivalent functions are said to be in the same equivalence class.
- All functions in an equivalence class have the same multiplicative complexity, i.e., multiplicative complexity is affine invariant.

Multiplicative Complexity of Boolean functions $n \leq 6$

Method

Exhaustively construct all Boolean circuit topologies with 1,2, 3, ... AND gates, and mark the Boolean functions that can be generated by the circuits until a function from each equivalence class is generated.

Multiplicative complexity is

- ≤ 3, for $n = 4$ (8 equivalence classes)
- ≤ 4, for $n = 5$ (48 equivalence classes)[3]
- ≤ 6, for $n = 6$ (150,357 equivalence classes)[4]

[3] M. Sonmez Turan, R. Peralta, *The Multiplicative Complexity of Boolean Functions on Four and Five Variables*, https://eprint.iacr.org/2015/848

[4] C. Calik and M. Sonmez Turan and R. Peralta, *The Multiplicative Complexity of 6-variable Boolean Functions*, https://eprint.iacr.org/2018/002

Multiplicative Complexity of Boolean functions $n \leq 6$

Method

Exhaustively construct all Boolean circuit topologies with 1,2, 3, ... AND gates, and mark the Boolean functions that can be generated by the circuits until a function from each equivalence class is generated.

Multiplicative complexity is

- ≤ 3, for $n = 4$ (8 equivalence classes)
- ≤ 4, for $n = 5$ (48 equivalence classes)[3]
- ≤ 6, for $n = 6$ (150,357 equivalence classes)[4]

How about for larger n? Vectorial Boolean functions? Multiplicative Complexity of AES S-box?

[3] M. Sonmez Turan, R. Peralta, *The Multiplicative Complexity of Boolean Functions on Four and Five Variables*, https://eprint.iacr.org/2015/848

[4] C. Calik and M. Sonmez Turan and R. Peralta, *The Multiplicative Complexity of 6-variable Boolean Functions*, https://eprint.iacr.org/2018/002

Thanks!

More information on Lightweight Cryptography Project available at
`https://www.nist.gov/programs-projects/lightweight-cryptography`

Subscribe to our mailing list: `lwc-forum@list.nist.gov`

Additional comments/questions?

Email the team at `lightweight-crypto@nist.gov`