# Benchmarking Round 2 Candidates on Microcontrollers

**Çağdaş Çalık**[*], Munawar Hasan[*], Jinkeon Kang[*]

NIST Lightweight Cryptography Workshop
October 19, 2020

[*] Guest Researcher, NIST

Updated on Nov 18, 2020

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

INFORMATION TECHNOLOGY LABORATORY

# Outline

- Motivation and Scope
- Implementations, Platforms, Test Cases
- The Benchmarking Framework
- Results
- Conclusion and Next Steps

Commercial equipment and software referred to in this paper are identified for informational purposes only and does not imply recommendation of or endorsement by the National Institute of Standards and Technology, nor does it imply that the products so identified are necessarily the best available for the purpose.

# Motivation and Scope

- Motivation
  - Evaluate the performance of Round 2 candidates on microcontrollers
  - Compare the candidates against existing NIST standards
- Goals
  - High coverage of implementations
  - Wide range of test cases
  - Verification of the implementations and results
- Scope
  - API compatible implementations
  - Official versions of the algorithms
  - This presentation: only primary variants

# Implementations

- Implementations were gathered from
  - Submission packages
  - Websites and GitHub repositories of the candidates
  - Third party software benchmarking projects

| Language* | AEAD (89 Variants) | Hash (19 Variants) | Total | AEAD-Primary (32 Variants) | Hash-Primary (12 Variants) |
|---|---|---|---|---|---|
| C | 161 | 41 | 202 | 66 | 27 |
| C / AVR | 73 | 15 | 88 | 26 | 10 |
| C / ARM / AVR | 9 | 4 | 13 | 3 | 2 |
| ARM | 35 | 4 | 39 | 18 | 4 |
| AVR | 19 | 16 | 35 | 7 | 7 |
| **Total** | **297** | **80** | **377** | **120** | **50** |

* Languages used in a single implementation folder

# Implementations - Remarks

- Primary variants of COMET*, ESTATE, mixFeed, and SAEAES* have only reference implementations.

- Problems faced in the benchmarking process
  - Build errors
  - Decryption failures
  - Program crash
  - Unsupported input sizes
  - Test vector verification failures

* Uses T-table based AES implementation.

# Platforms

| Board | Microcontroller / Core | Frequency | Flash | SRAM |
|---|---|---|---|---|
| Arduino Uno Rev3 | ATmega328P - AVR (8-bit) | 16 MHz | 32 KB | 2 KB |
| Arduino Nano Every | ATMega4809 - AVR (8-bit) | 20 MHz | 48 KB | 6 KB |
| Arduino MKR Zero | SAMD21 - ARM Cortex-M0+ (32-bit) | 48 MHz | 256 KB | 32 KB |
| Arduino Due | AT91SAM3X8E - ARM Cortex-M3 (32-bit) | 84 MHz | 512 KB | 96 KB |
| Arduino Nano 33 BLE | nRF52840 - ARM Cortex-M4F (32-bit) | 64 MHz | 1 MB | 256 KB |
| HiFive1 Rev B | SiFive FE310-G002 - RV32 IMAC (32-bit) | 320 MHz | 4 MB | 16 KB |

# Platforms

| Board | Microcontroller / Core | Frequency | Flash | SRAM |
|---|---|---|---|---|
| Arduino Uno Rev3 | ATmega328P - AVR (8-bit) | 16 MHz | 32 KB | 2 KB |
| Arduino Nano Every | ATMega4809 - AVR (8-bit) | 20 MHz | 48 KB | 6 KB |
| Arduino MKR Zero | SAMD21 - ARM Cortex-M0+ (32-bit) | 48 MHz | 256 KB | 32 KB |
| Arduino Due | AT91SAM3X8E - ARM Cortex-M3 (32-bit) | 84 MHz | 512 KB | 96 KB |
| Arduino Nano 33 BLE | nRF52840 - ARM Cortex-M4F (32-bit) | 64 MHz | 1 MB | 256 KB |
| HiFive1 Rev B | SiFive FE310-G002 - RV32 IMAC (32-bit) | 320 MHz | 4 MB | 16 KB |

# Test Cases

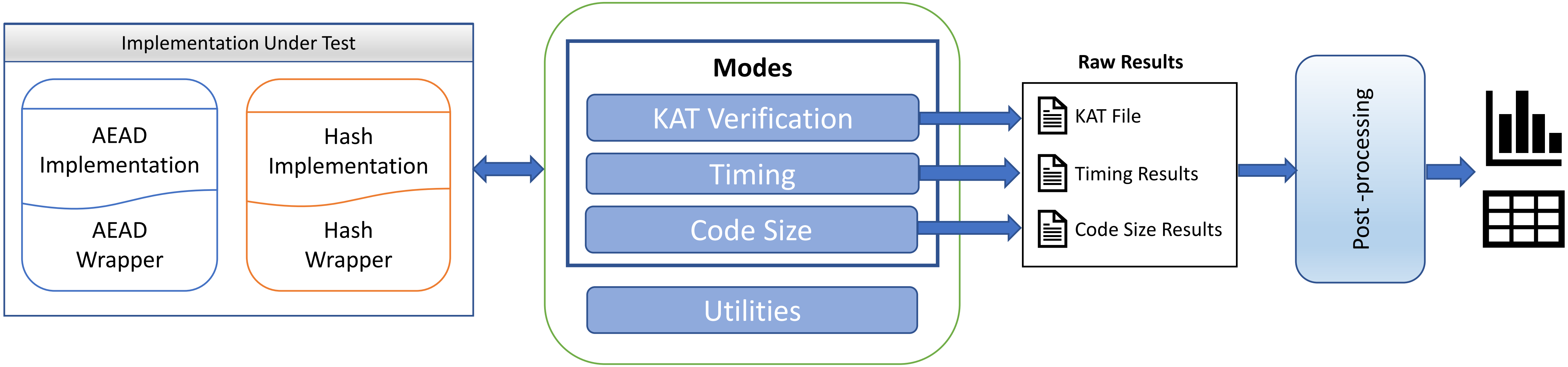
- Time
  - Varying input sizes for *Plaintext* and *Associated Data* for AEAD and Hash functions
    - Short inputs: ranging from 0 to 128 bytes
    - Long inputs*: ranging from 256 to 2048 bytes with 128-byte increments

- Size
  - For AEAD algorithms, the sizes for *enc-only*, *dec-only* implementations are calculated.
  - Actual sizes are calculated by taking the difference w.r.t. the *empty cipher*.

*Omitted for AVR.

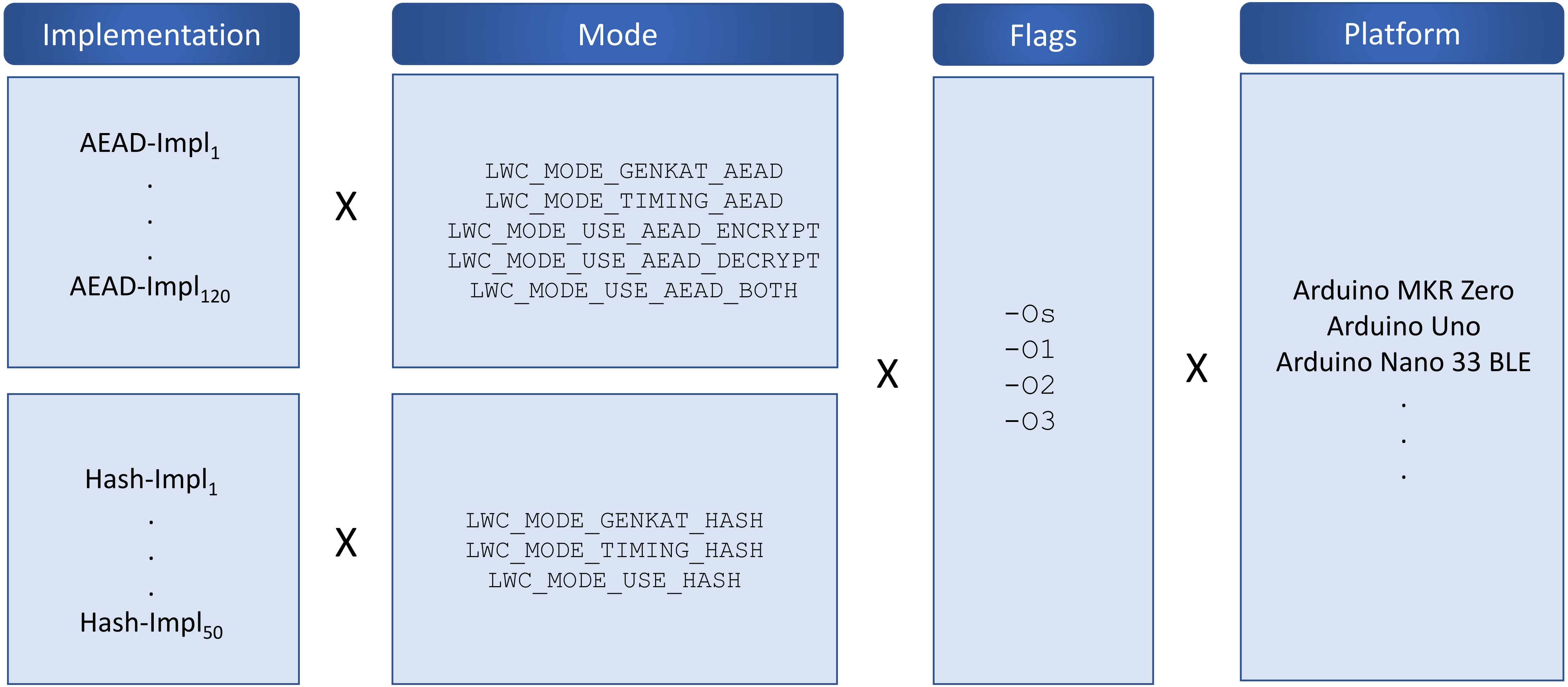
# The Benchmarking Framework

- The framework consists of C++ code for carrying out the experiments, and scripts for automating the build process and post-processing the results.
- It uses the PlatformIO embedded development platform and the GNU toolchains[1,2,3].



[1] (GNU Tools for Arm Embedded Processors 7-2017-q4-major) 7.2.1 20170904 (release) [ARM/embedded-7-branch revision 255204]
[2] (GNU MCU Eclipse ARM Embedded GCC, 64-bit) 8.2.1 20181213 (release) [gcc-8-branch revision 267074]
[3] (AVR_8_bit_GNU_Toolchain_3.6.2_1759) 5.4.0

# Build Configurations



Implementation

AEAD-Impl$_1$

.

.

.

AEAD-Impl$_{120}$

Hash-Impl$_1$

.

.

.

Hash-Impl$_{50}$

X

Mode

```
LWC_MODE_GENKAT_AEAD
LWC_MODE_TIMING_AEAD
LWC_MODE_USE_AEAD_ENCRYPT
LWC_MODE_USE_AEAD_DECRYPT
LWC_MODE_USE_AEAD_BOTH
```

```
LWC_MODE_GENKAT_HASH
LWC_MODE_TIMING_HASH
LWC_MODE_USE_HASH
```

X

Flags

```
-Os
-O1
-O2
-O3
```

X

Platform

Arduino MKR Zero

Arduino Uno

Arduino Nano 33 BLE

.

.

.

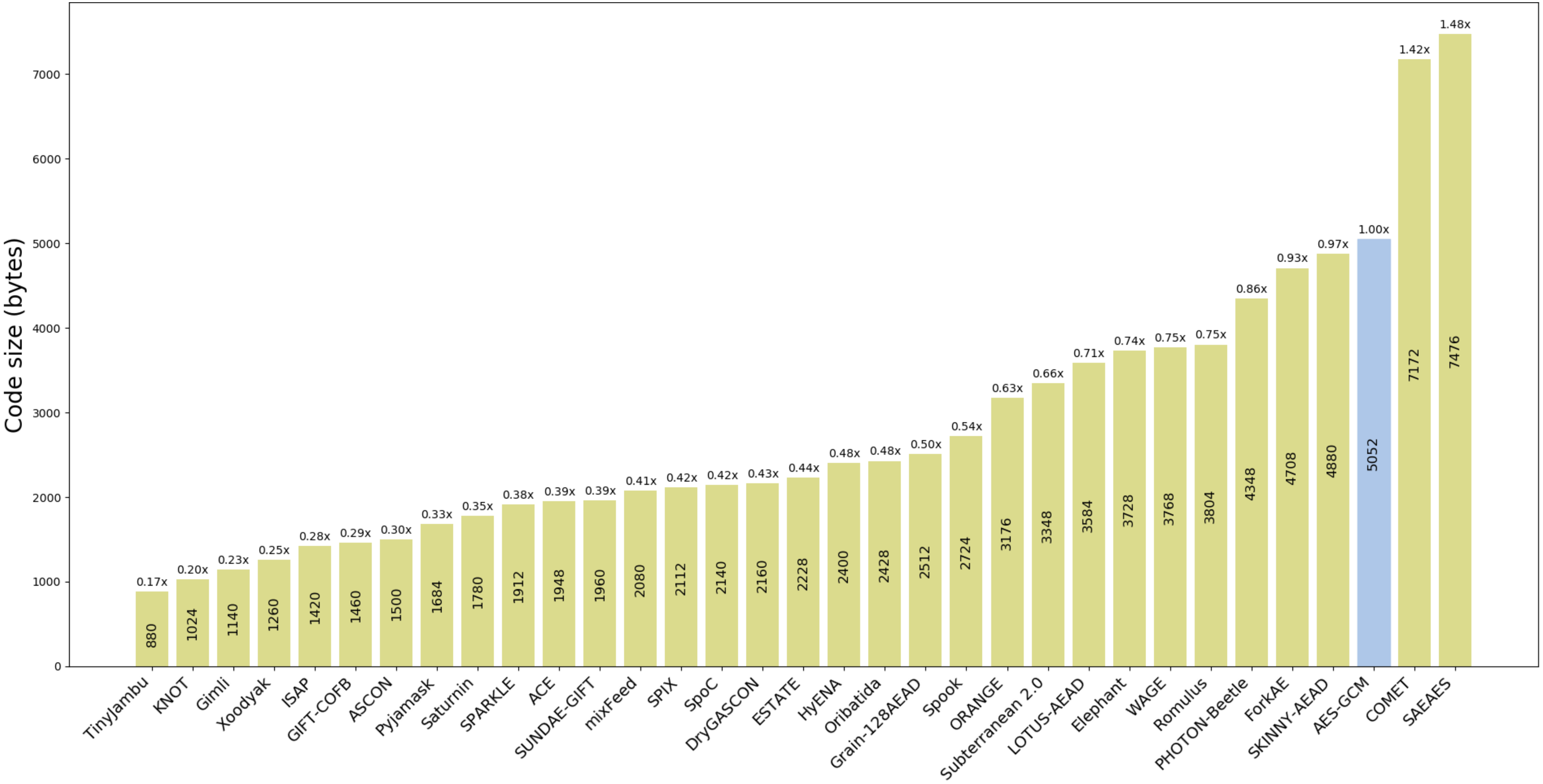
# Benchmark Results

- Code size (AEAD & Hash)
- Timing (AEAD & Hash)
- Pairwise-comparison of AEAD algorithms against AES-GCM (AEAD only)

- Benchmarks include AES-GCM and SHA-256 from Mbed-TLS library for comparison.

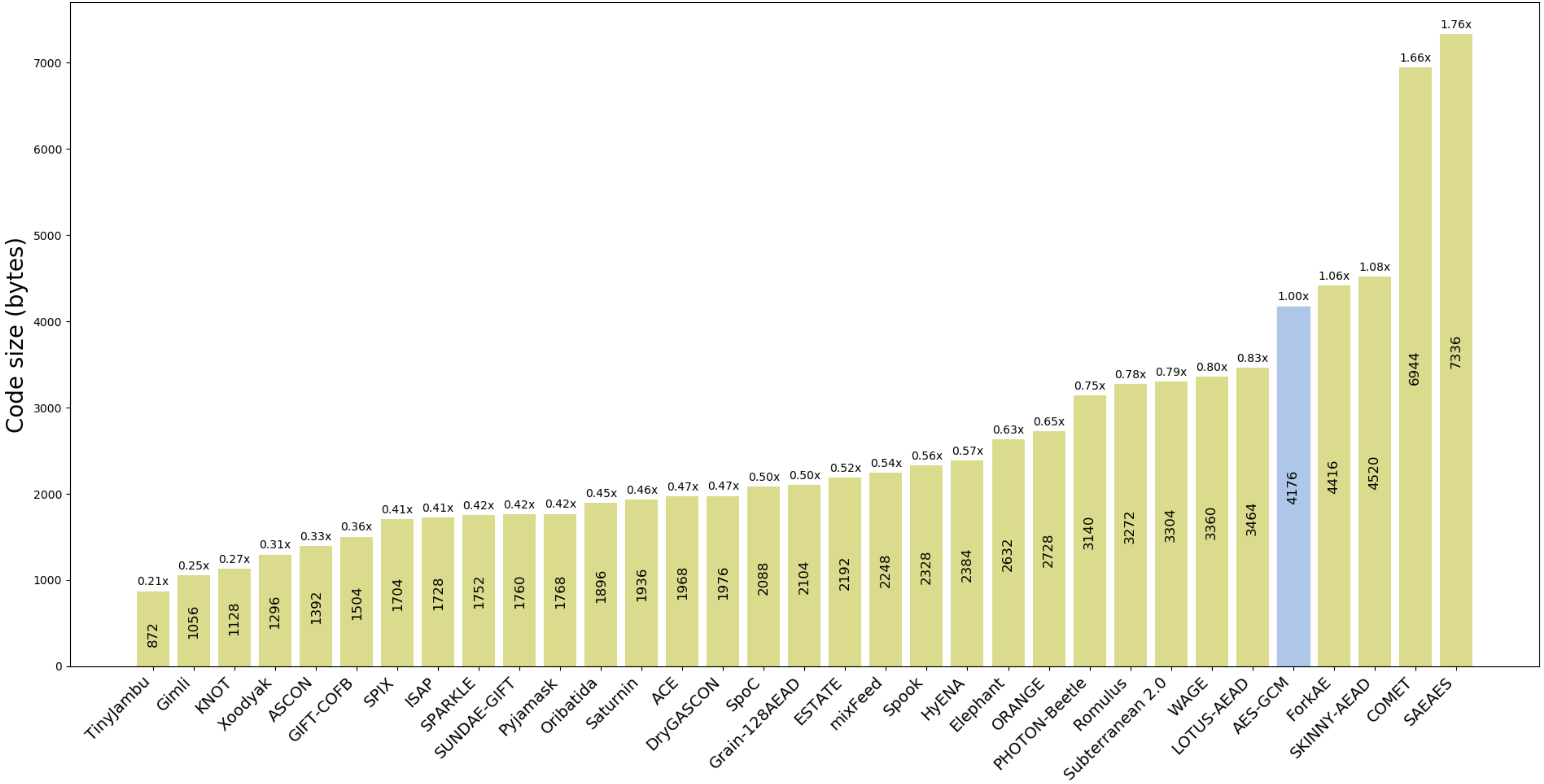
# Code Size for Primary AEAD Variants* on Cortex-M0+

* Smallest sized implementation of each variant among all its implementations compiled with four different optimization flags



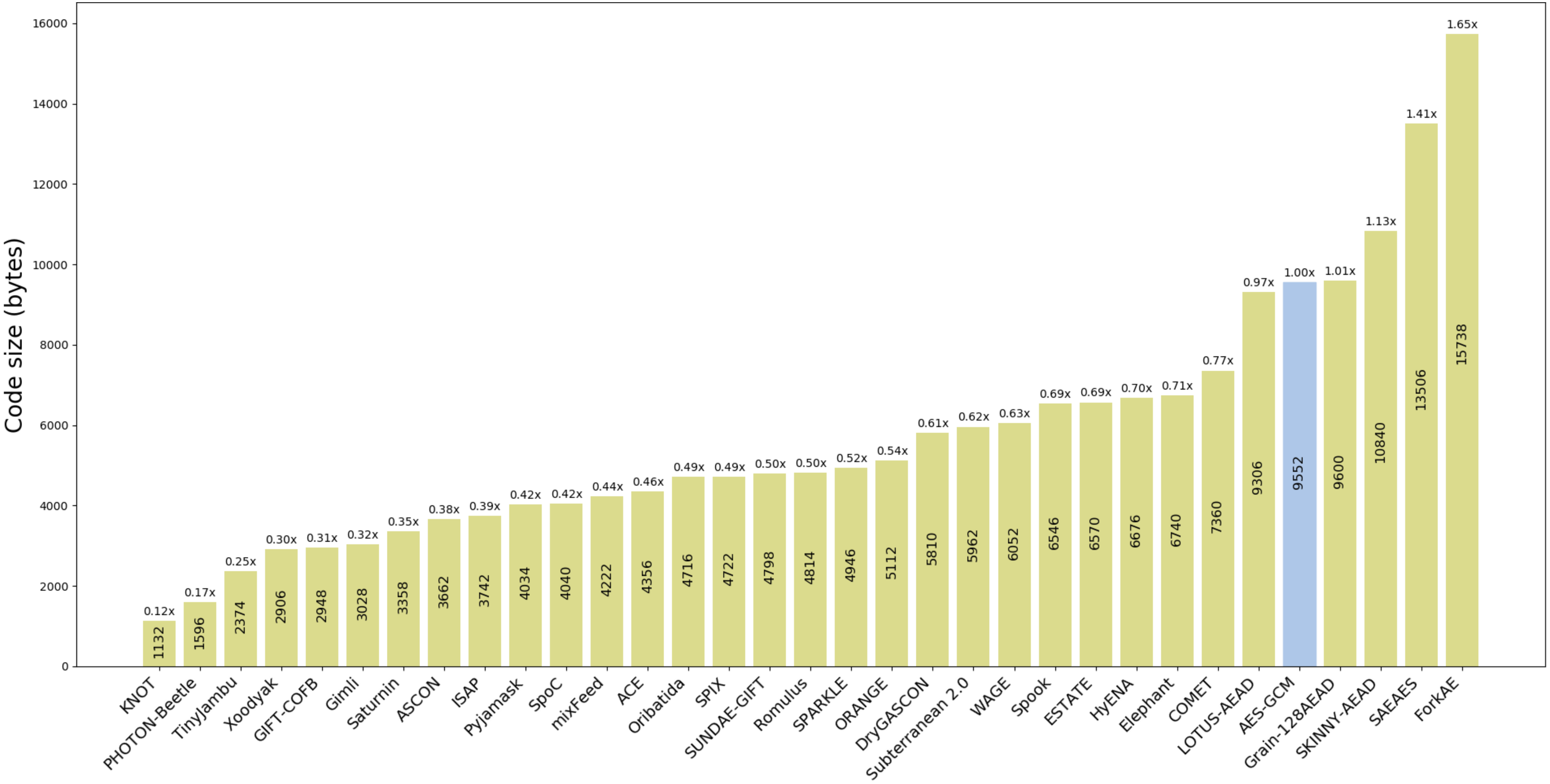
# Code Size for Primary AEAD Variants* on Cortex-M4F

* Smallest sized implementation of each variant among all its implementations compiled with four different optimization flags



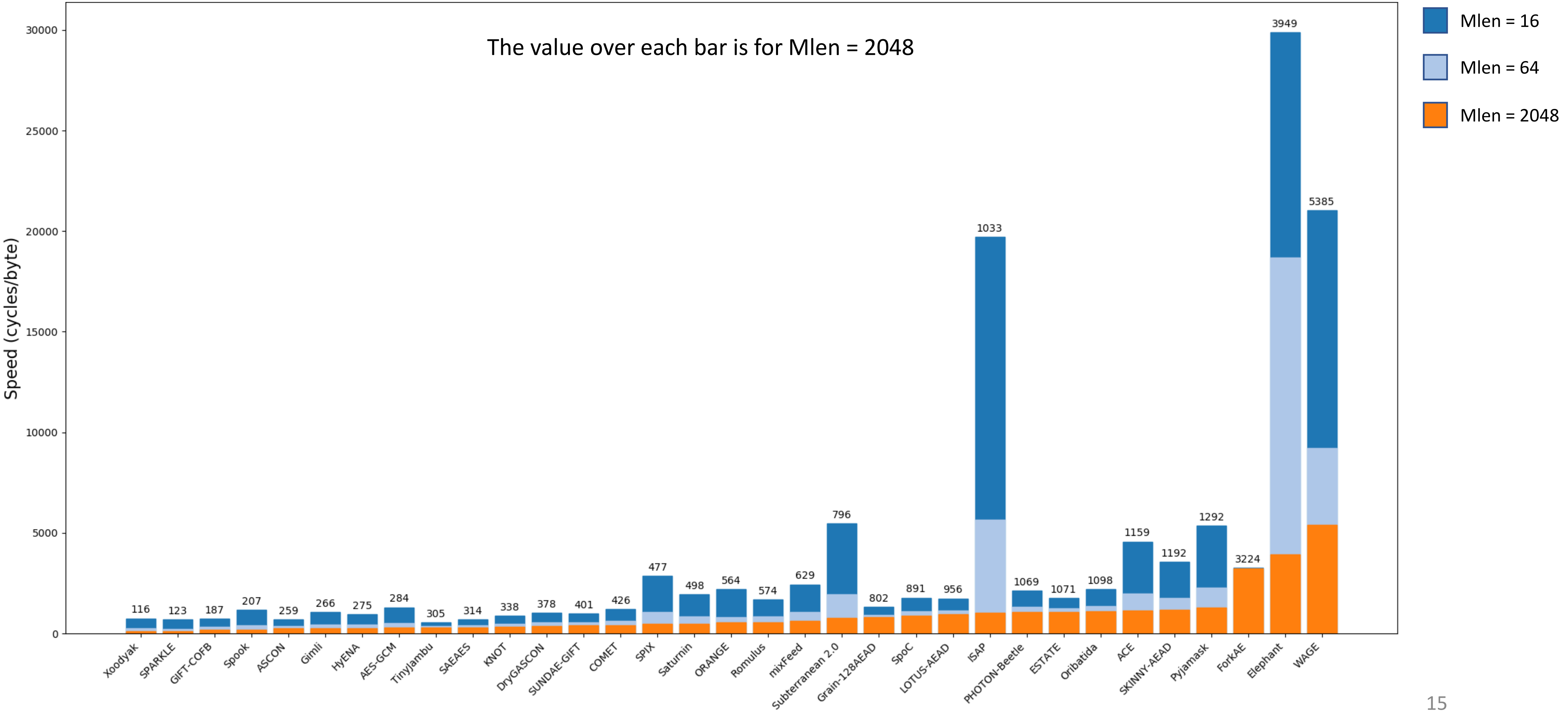
# Code Size for Primary AEAD Variants* on AVR

* Smallest sized implementation of each variant among all its implementations compiled with four different optimization flags



# Timings for Primary AEAD Variants[*] on Cortex-M0+ (AD Length=0, Msg Length=16, 64, 2048)

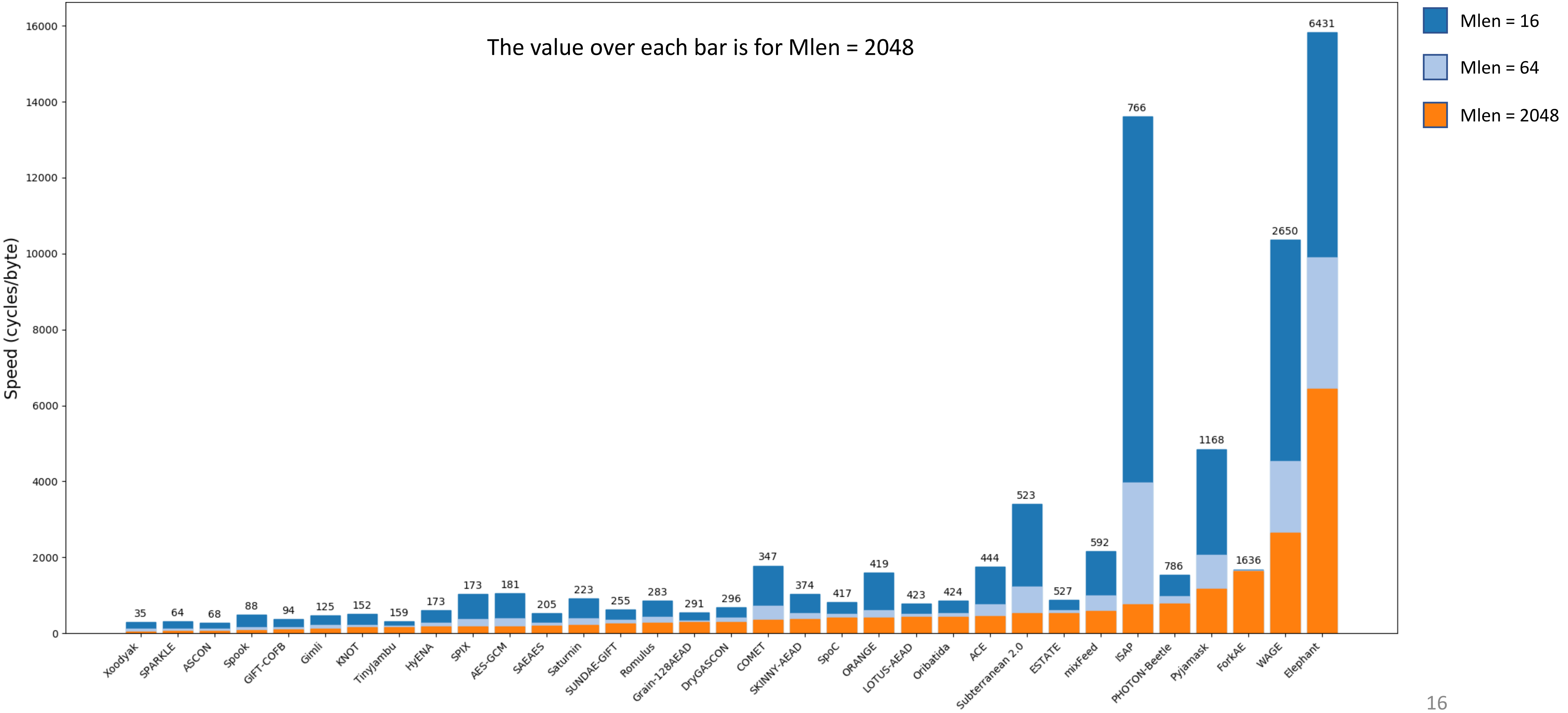
[*] Fastest implementation of each variant among all its implementations compiled with four different optimization flags



The value over each bar is for Mlen = 2048

# Timings for Primary AEAD Variants* on Cortex-M4F (AD Length=0, Msg Length=16, 64, 2048)

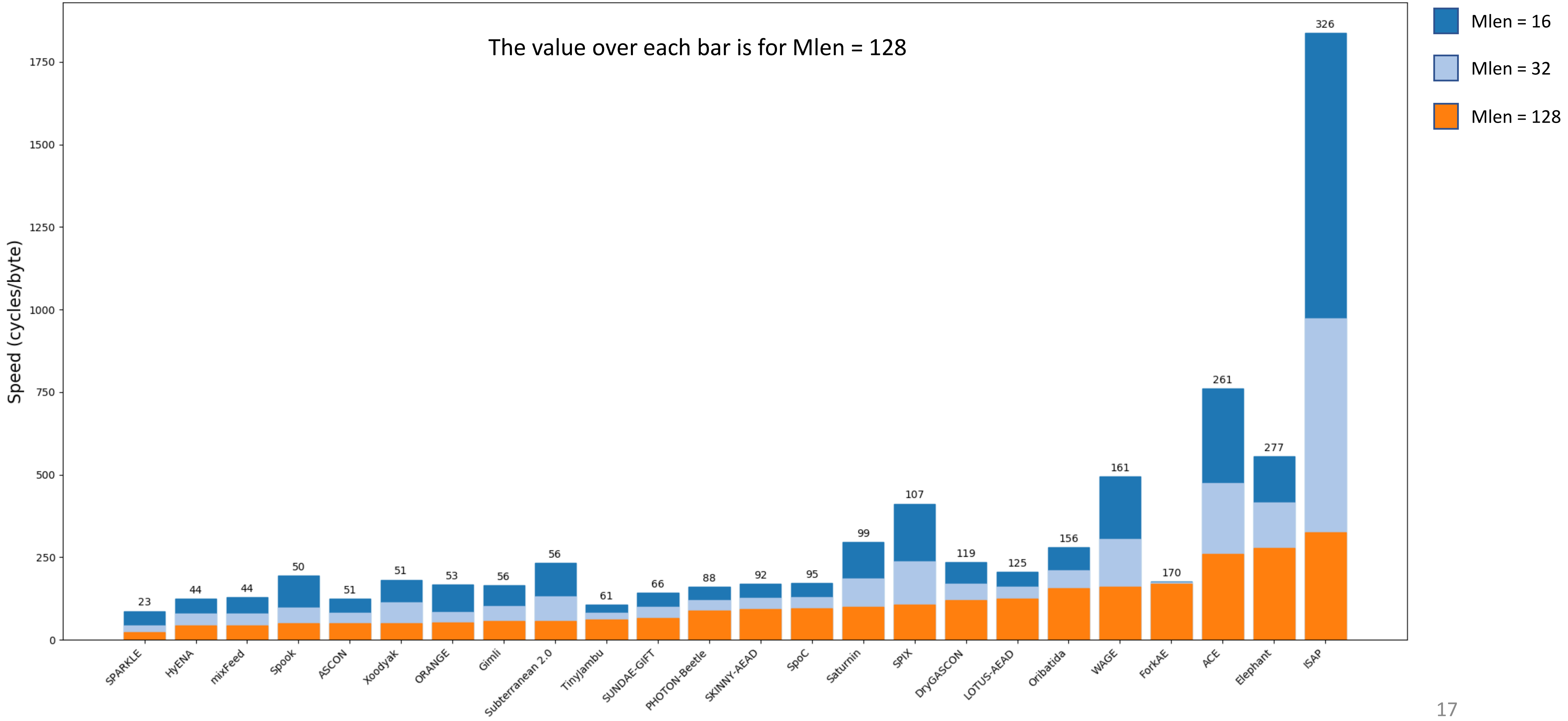
* Fastest implementation of each variant among all its implementations compiled with four different optimization flags



The value over each bar is for Mlen = 2048

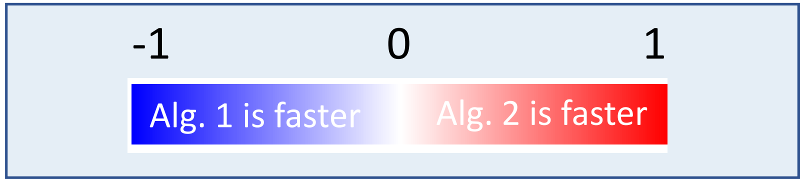
# Timings for Primary AEAD Variants* on AVR (AD Length=0, Msg Length=16, 32, 128)

* Fastest implementation of each variant among all its implementations compiled with four different optimization flags


The value over each bar is for Mlen = 128

# Pairwise Comparison of AEAD Algorithms

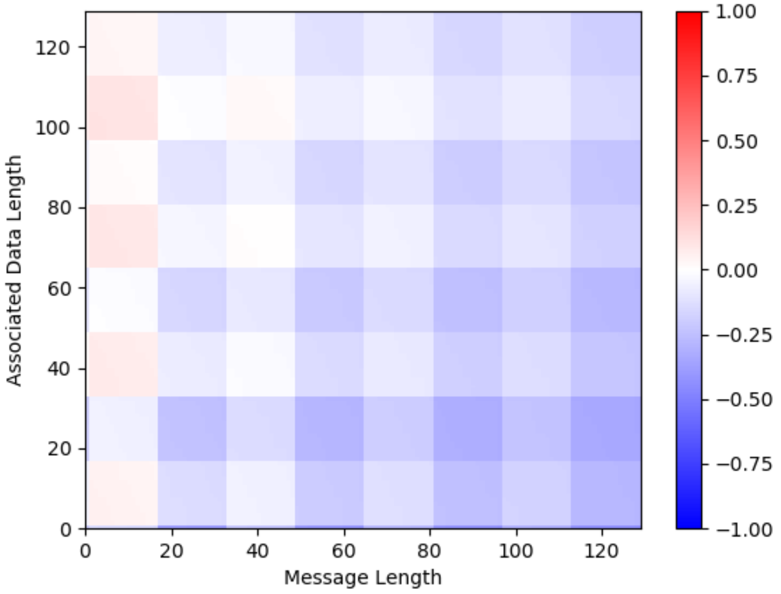
- A 2D plot is created by comparing the execution times of two AEAD algorithms for each *Plaintext* and *Associated Data* length from 0 to 128 bytes.

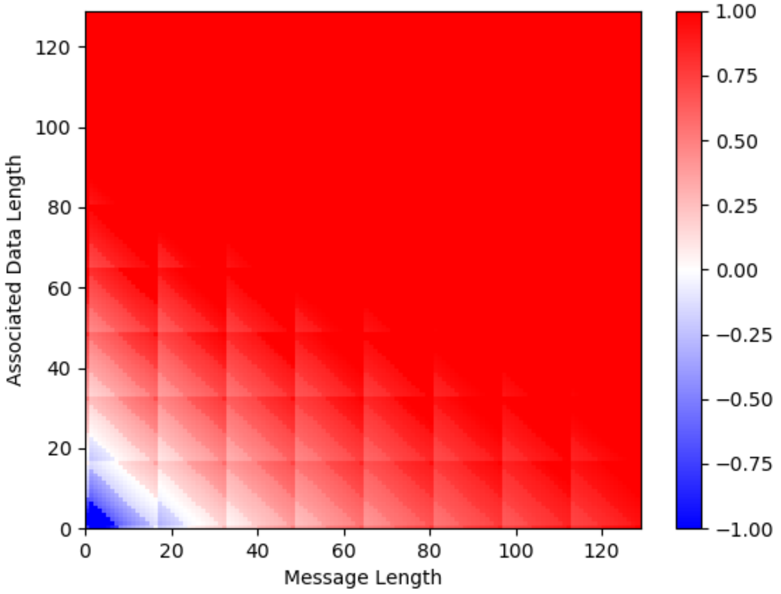


-1 0 1

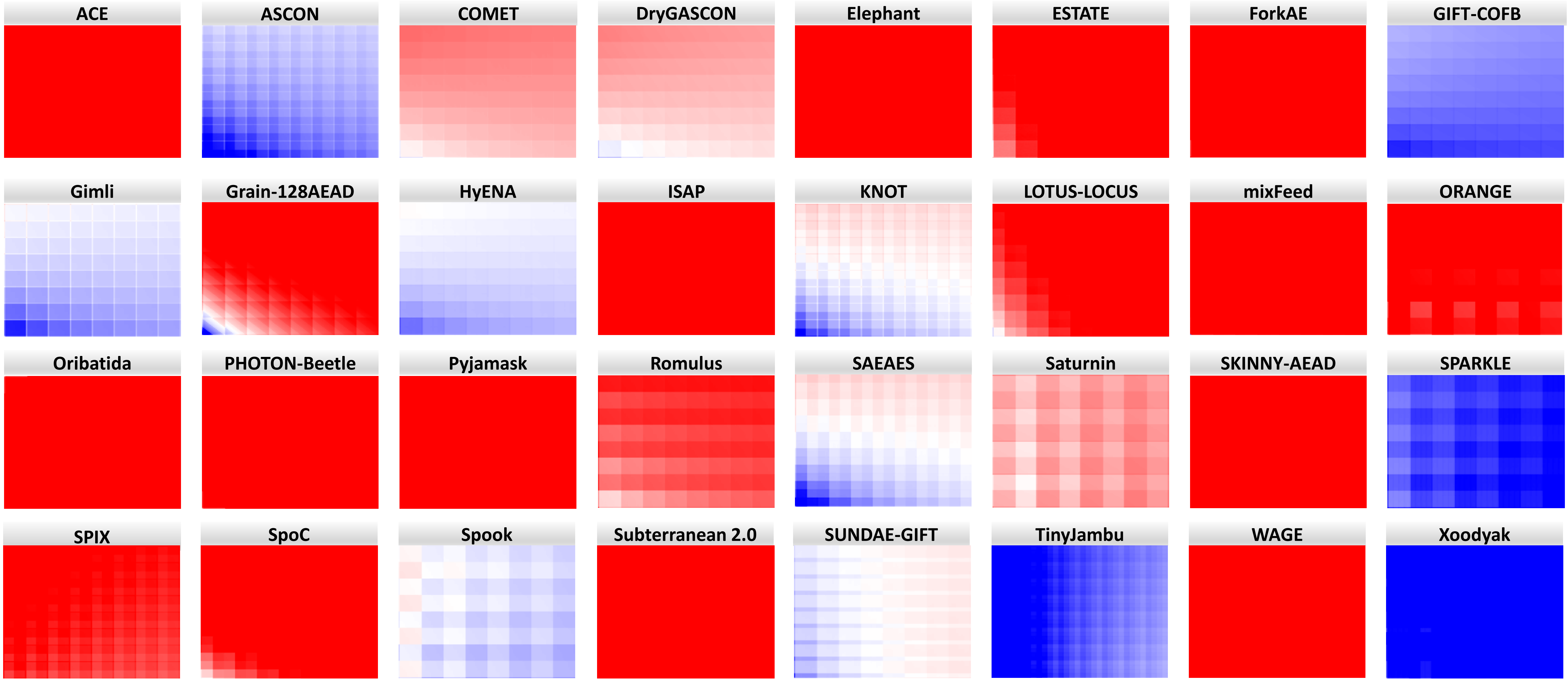
Alg. 1 is faster

Alg. 2 is faster



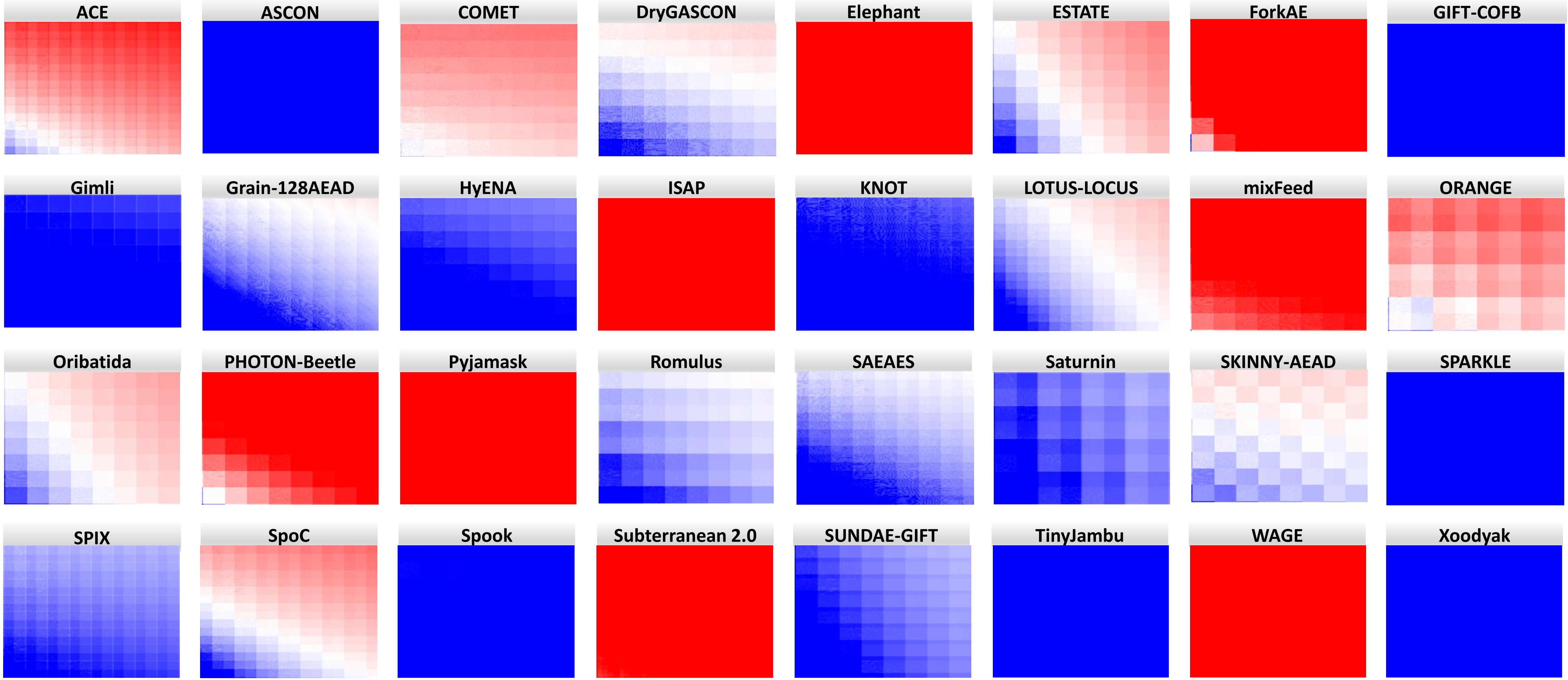


# Primary AEAD Variants[1] v. AES-GCM[2] on Cortex-M0+



[1] The fastest implementation of each variant. [2] Mbed TLS implementation with `MBEDTLS_AES_ROM_TABLES` and `MBEDTLS_AES_FEWER_TABLES` defined.

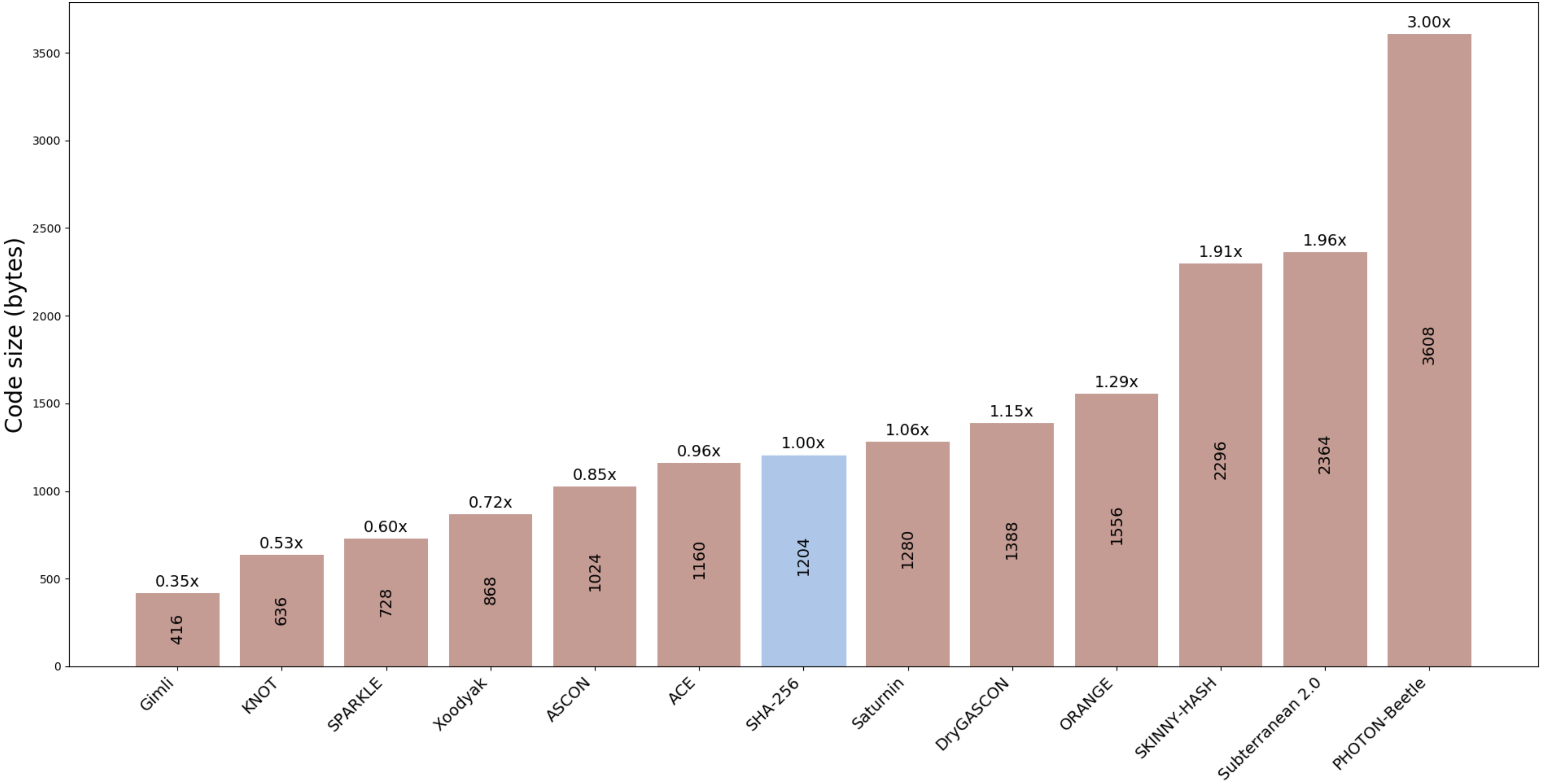
# Primary AEAD Variants[1] v. AES-GCM[2] on Cortex-M4F



[1] The fastest implementation of each variant. [2] Mbed TLS implementation with `MBEDTLS_AES_ROM_TABLES` and `MBEDTLS_AES_FEWER_TABLES` defined.

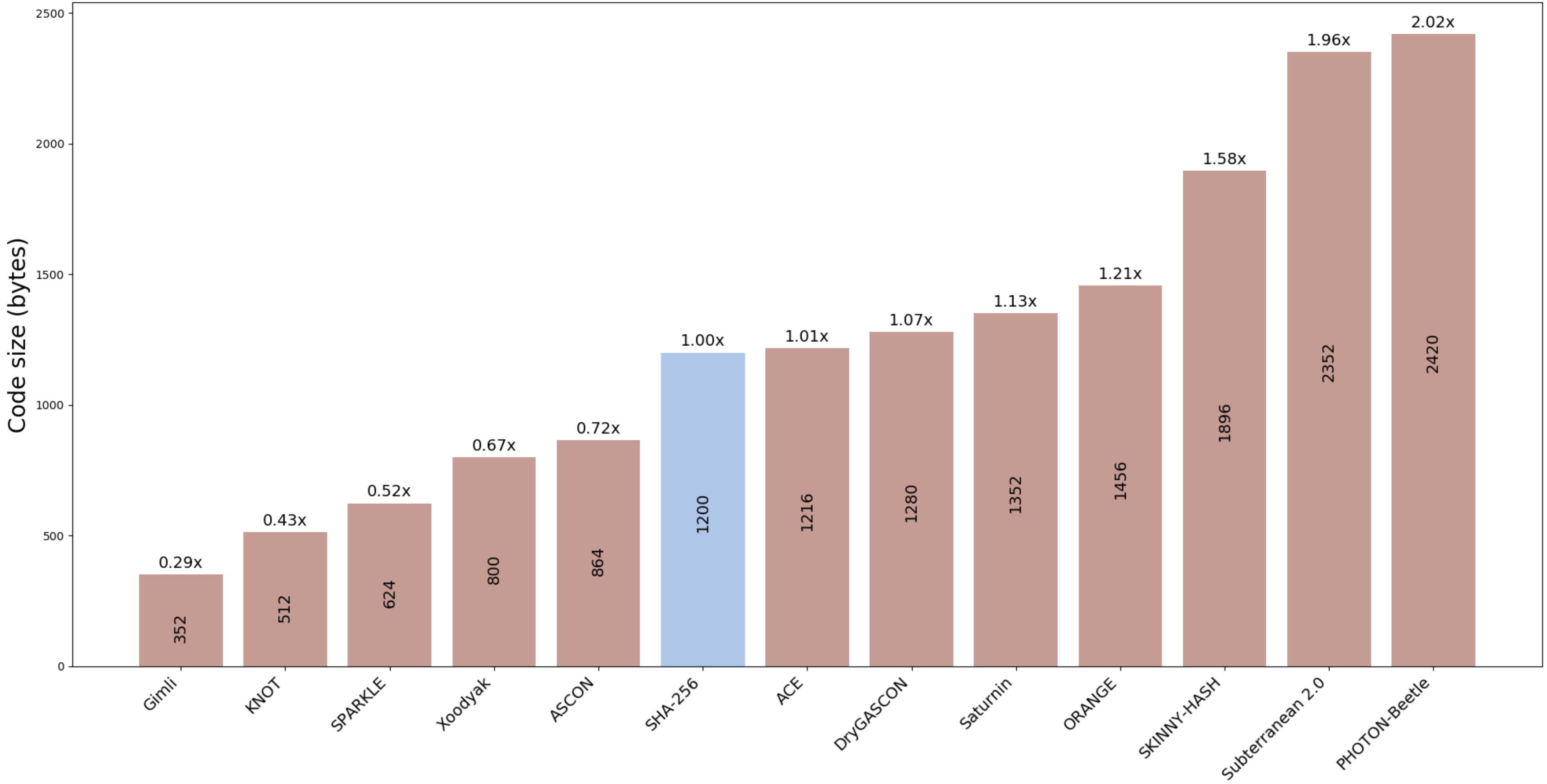
# Code Size for Primary Hash Variants* on Cortex-M0+

* Smallest sized implementation of each variant among all its implementations compiled with four different optimization flags.



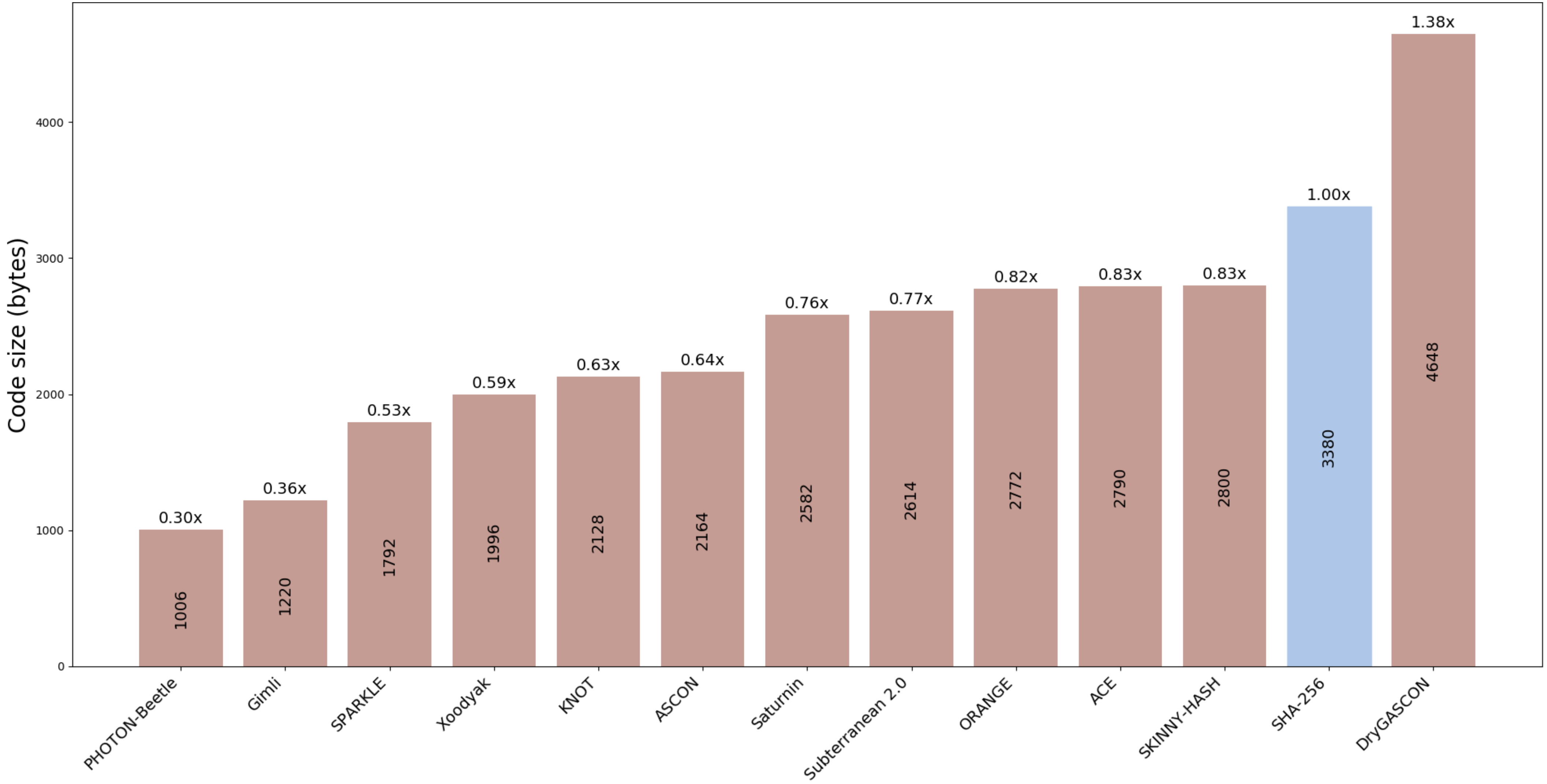
# Code Size for Primary Hash Variants* on Cortex-M4F

* Smallest sized implementation of each variant among all its implementations compiled with four different optimization flags.



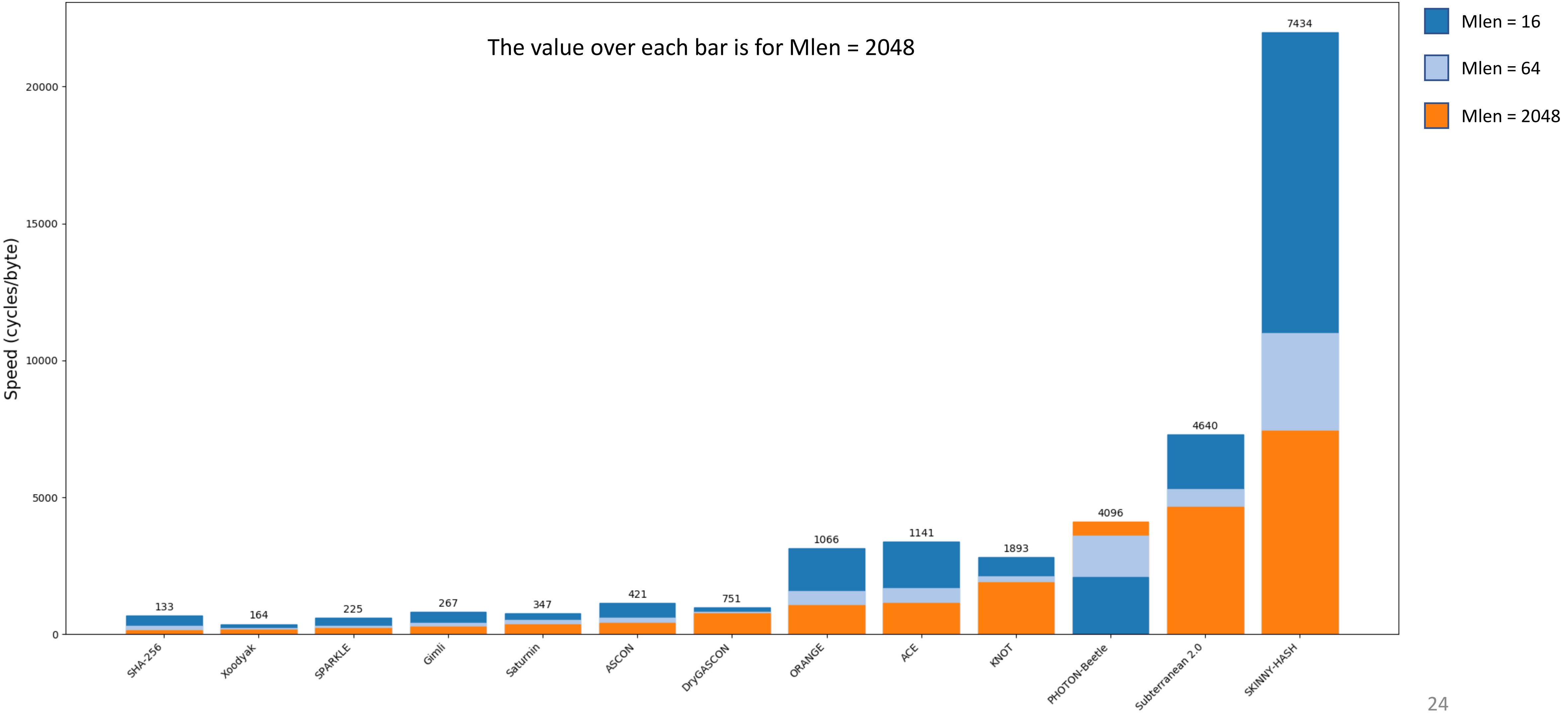
# Code Size for Primary Hash Variants* Variants on AVR

* Smallest sized implementation of each variant among all its implementations compiled with four different optimization flags.



# Timings for Primary Hash Variants* on Cortex-M0+ (Msg Length=16, 64, 2048)

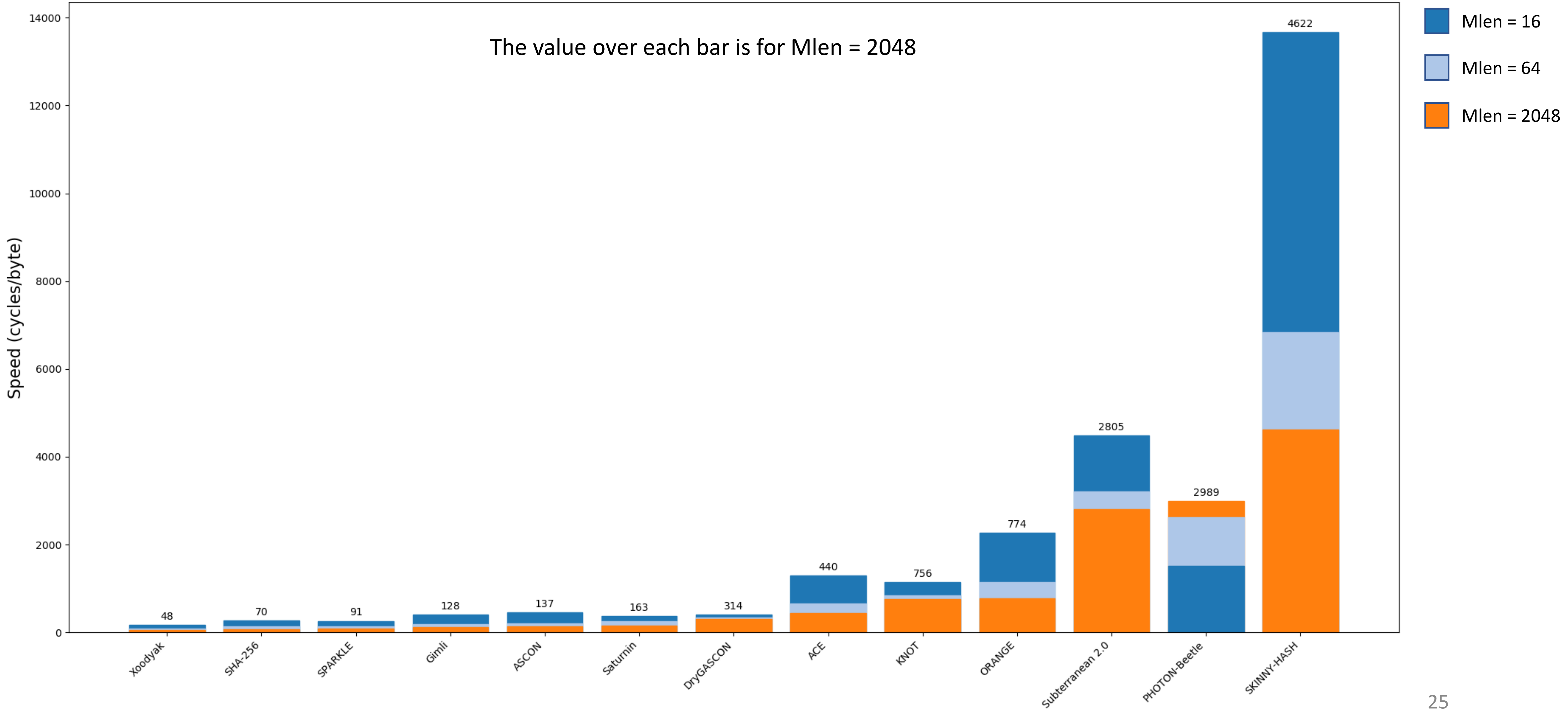
* Fastest implementation of each variant among all its implementations compiled with four different optimization flags



The value over each bar is for Mlen = 2048

# Timings for Primary Hash Variants* on Cortex-M4F (Msg Length=16, 64, 2048)

* Fastest implementation of each variant among all its implementations compiled with four different optimization flags



The value over each bar is for Mlen = 2048

# Summary of Results

Benchmarking is challenging, due to the range of platforms, implementation tradeoffs, and different use cases.

**AEAD**

- Most of the candidates achieved smaller *code size* compared to AES-GCM*.
- On ARM Cortex-M0+ about half of the candidates and on ARM Cortex-M4F most candidates showed performance improvement at least in some of the test cases over AES-GCM* .

**Hash**

- Depending on the platform, four to seven candidates had smaller code size than SHA-256*.
- Most of the candidates performed worse compared to SHA-256 in timing experiments.

* Mbed TLS implementation.

# Conclusion and Next Steps

- Fair and extensive performance evaluation of the candidates on microcontrollers will contribute to the selection of the finalists.
- The benchmark results published by the submitters, third party benchmarking projects, and academic papers are also taken into consideration in the evaluation process.
- The benchmarking framework and the results will be available at: https://github.com/usnistgov/Lightweight-Cryptography-Benchmarking

- Next Steps
  - Keep the implementation database up to date
  - Resolve the issues for implementations where benchmarking could not be performed
  - Add new platforms
  - Verify and publish the results

# Contact

**Project webpage:** https://csrc.nist.gov/projects/lightweight-cryptography

**GitHub:** https://github.com/usnistgov/Lightweight-Cryptography-Benchmarking

**Forum:** lwc-forum@list.nist.gov

**Contact email:** lightweight-crypto@nist.gov