Software Benchmarking of the 2\textsuperscript{nd} round CAESAR Candidates

Ralph Ankele\textsuperscript{1}, Robin Ankele\textsuperscript{2}

\textsuperscript{1}Royal Holloway, University of London, UK \textsuperscript{2}University of Oxford, UK

September 27, 2016

Directions in Authenticated Ciphers - Nagoya, Japan
Motivation

Use Case 1: Lightweight applications (resource constrained environments)

Use Case 2: High-performance applications

- critical: efficiency on 64-bit CPUs (servers) and/or dedicated hardware
- desirable: efficiency on 32-bit CPUs (small smartphones)
- desirable: constant time when the message length is constant
- message sizes: usually long (more than 1024 bytes), sometimes shorter

Use Case 3: Defense in depth

\(^1\)CAESAR usecases on CAESAR mailing list (16. July 2016) by Dan J. Bernstein: https://groups.google.com/forum/#!topic/crypto-competitions/DLv193SPSDc
Overview

1. Classification of the 2\textsuperscript{nd} round CAESAR Candidates

2. Software Optimizations

3. Benchmarking Framework

4. Results

5. Conclusions
1. Classification of the 2nd round CAESAR Candidates

2. Software Optimizations

3. Benchmarking Framework

4. Results

5. Conclusions
CAESAR Round 2 candidates

<table>
<thead>
<tr>
<th>ACORN</th>
<th>AEGIS</th>
<th>AES-COPA</th>
<th>AES-JAMBU</th>
<th>AES-OTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEZ</td>
<td>Ascon</td>
<td>CLOC</td>
<td>Deoxys</td>
<td>ELMd</td>
</tr>
<tr>
<td>HS1-SIV</td>
<td>ICEPOLE</td>
<td>Joltik</td>
<td>Ketje</td>
<td>Keyak</td>
</tr>
<tr>
<td>MORUS</td>
<td>Minalpher</td>
<td>NORX</td>
<td>OCB</td>
<td>OMD</td>
</tr>
<tr>
<td>PAEQ</td>
<td>POET</td>
<td>PRIMATEs</td>
<td>SCREAM</td>
<td>SHELL</td>
</tr>
<tr>
<td>SILC</td>
<td>STRIBOB</td>
<td>Tiaoxin</td>
<td>TriviA-ck</td>
<td>$\pi$-Cipher</td>
</tr>
</tbody>
</table>

Ralph Ankele - Royal Holloway, University of London
Software Benchmarking of the 2nd round CAESAR Candidates
Type

- Block Cipher: 15
- Compression Function: 1
- Permutations: 2
- Stream Cipher: 4
- Sponge Construction: 8
Underlying Primitive

- AES: 10
- Keccak: 3
- SPN: 3
- AES Round: 3
- Dedicated Block Cipher: 1
- Dedicated Stream Cipher: 1
- Dedicated Permutation: 1
- SHA2: 1
- ARX: 1
- LRX: 1
- Others: 3

Ralph Ankele - Royal Holloway, University of London
Software Benchmarking of the 2nd round CAESAR Candidates
Parallel Encryption/Decryption

- Fully/Fully: 14
- Fully/No: 10
- No/No: 5
- Partly/Partly: 1
Encryption of a message block $M_i$ only depends on message blocks $M_1 \ldots M_{i-1}$. 
Inverse Free

- Yes: 19
- No: 10
Nonce-Missuse Resistance

Longest common prefix: an adversary can observe the longest common prefix of messages for repeated nonces

Max: the repetition of nonces only leak the ability to see a repeated message
1. Classification of the 2\textsuperscript{nd} round CAESAR Candidates

2. Software Optimizations

3. Benchmarking Framework

4. Results

5. Conclusions
Software Optimizations

- Streaming SIMD Extensions: 9
- AES New Instructions: 12
- No Software Optimization: 6
- Advanced Vector Instructions: 7
- NEON: 4
- Dedicated Processor Optimizations: 4

Ralph Ankele - Royal Holloway, University of London
Software Benchmarking of the 2nd round CAESAR Candidates
AES-New Instructions

Instructions

- Introduced with Intel® 2010 Westmere microarchitecture
- Consists of 6 new instructions that are implemented in hardware
- Four instructions for encryption/decryption (i.e. AESENC, AESENCLAST, AESDEC, AESDECLAST)
- Two instructions for the keyschedule (i.e. AESKEYGENASSIST, AESIMC)

Performance

- 10 times faster for parallel modes (i.e. CTR)
- 2-3 times faster for non-parallel modes (i.e. CBC)

Security

- Improved security against side channel attacks [Gue12]
AES-New Instructions

Instructions

- Introduced with Intel® 2010 Westmere microarchitecture
- Consists of 6 new instructions that are implemented in hardware
- Four instructions for encryption/decryption (i.e. AESENC, AESENCLAST, AESDEC, AESDECLAST)
- Two instructions for the keyschedule (i.e. AESKEYGENASSIST, AESIMC)

Performance

- 10 times faster for parallel modes (i.e. CTR)
- 2-3 times faster for non-parallel modes (i.e. CBC)

Security

- Improved security against side channel attacks [Gue12]
AES-New Instructions

Instructions

- Introduced with Intel® 2010 Westmere microarchitecture
- Consists of 6 new instructions that are implemented in hardware
- Four instructions for encryption/decryption (i.e. AESENC, AESENCLAST, AESDEC, AESDECLAST)
- Two instructions for the keyschedule (i.e. AESKEYGENASSIST, AESIMC)

Performance

- 10 times faster for parallel modes (i.e. CTR)
- 2-3 times faster for non-parallel modes (i.e. CBC)

Security

- Improved security against side channel attacks [Gue12]
Streaming SIMD Extensions

**Instructions**

- Vector-mode operations that enables parallel execution of one instruction on multiple data
- 16 · 128-bit registers (xmm0-15)
- Expanded over Intel® processor generations to include SSE2, SSE3/SSE3S and SSE4

Image: https://software.intel.com/sites/default/files/37208.gif
Advanced Vector Extensions

Instructions

- Introduced with Intel® SandyBridge microarchitecture
- Extends SSE 128-bit registers with 16 new 256-bit registers (ymm0-15)
- Support of three-operand non-destructive operations (two-operand instructions *e.g.* $A = A + B$ are replaced by three-operand instructions *e.g.* $A = B + C$)
- AVX2 instructions expand integer vector types and vector shift operations

Performance

- AVX is 1.8 times faster than fastest SSE4.2 instructions [Len14]
- AVX2 is 2.8 times faster than fastest SSE4.2 instructions [Len14]
Advanced Vector Extensions

Instructions

- Introduced with Intel® SandyBridge microarchitecture
- Extends SSE 128-bit registers with 16 new 256-bit registers (ymm0-15)
- Support of three-operand non-destructive operations (two-operand instructions e.g. $A = A + B$ are replaced by three-operand instructions e.g. $A = B + C$)
- AVX2 instructions expand integer vector types and vector shift operations

Performance

- AVX is 1.8 times faster than fastest SSE4.2 instructions [Len14]
- AVX2 is 2.8 times faster than fastest SSE4.2 instructions [Len14]
Instructions

- Advanced SIMD instructions for ARM processors available since CORTEX-A microarchitecture
- 32 · 64-bit registers (dual view 16 · 128-bit registers)

Performance

- 2-8 times performance boost [neo]

Image: http://www.arm.com/assets/images/NEON_ISA.jpg
Instructions

- Advanced SIMD instructions for ARM processors available since CORTEX-A microarchitecture
- 32 · 64-bit registers (dual view 16 · 128-bit registers)

Performance

- 2-8 times performance boost [neo]

Image: http://www.arm.com/assets/images/NEON_ISA.jpg
1. Classification of the 2nd round CAESAR Candidates

2. Software Optimizations

3. Benchmarking Framework

4. Results

5. Conclusions
High Resolution Methods for CPU Timing Information

High Resolution Timers

- HPET (High Precision Event Timer)
- QueryPerformanceCounter
- `time()` and `clock()` posix functions
- TSC (Timer Stamp Counter)

Timer Stamp Counter

- 64-bit Machine State Register containing the number of cycles since last reset
- RDTSC instruction to read out
- Use CPUID instruction against out-of-order execution
- Our framework uses RDTSCP [Pao10] which is an optimised RDTSC + CPUID
High Resolution Methods for CPU Timing Information

High Resolution Timers

- HPET (High Precision Event Timer)
- QueryPerformanceCounter
- `time()` and `clock()` POSIX functions
- TSC (Timer Stamp Counter)

Timer Stamp Counter

- 64-bit Machine State Register containing the number of cycles since last reset
- RDTSC instruction to read out
- Use CPUID instruction against out-of-order execution
- Our framework uses RDTSCP [Pao10] which is an optimised RDTSC + CPUID
Benchmarking Framework

SUPERCOP [Ber16]

- System for Unified Performance Evaluation Related to Cryptographic Operations and Primitives
- Uses Timer Stamp Counter as Timer (with RDTSC)

BRUTUS [Saa16]

- Small codebase, rapid testing cycle
- Uses clock() as Timer

Our Framework

- Simple with only focus on Authenticated Encryption schemes
- Optimized Timer Stamp Counter (i.e. RDTSCP) for accurate timing measurements [Pao10]
- Reduction of noise using single user mode, averaging and median
Benchmarking Framework

SUPERCOP [Ber16]
- System for Unified Performance Evaluation Related to Cryptographic Operations and Primitives
- Uses Timer Stamp Counter as Timer (with RDTSC)

BRUTUS [Saa16]
- Small codebase, rapid testing cycle
- Uses clock() as Timer

Our Framework
- Simple with only focus on Authenticated Encryption schemes
- Optimized Timer Stamp Counter (i.e. RDTSCP) for accurate timing measurements [Pao10]
- Reduction of noise using single user mode, averaging and median
Benchmarking Framework

SUPERCOP [Ber16]

- System for Unified Performance Evaluation Related to Cryptographic Operations and Primitives
- Uses Timer Stamp Counter as Timer (with RDTSC)

BRUTUS [Saa16]

- Small codebase, rapid testing cycle
- Uses clock() as Timer

Our Framework

- Simple with only focus on Authenticated Encryption schemes
- Optimized Timer Stamp Counter (i.e. RDTSCP) for accurate timing measurements [Pao10]
- Reduction of noise using single user mode, averaging and median
Measurement Setup

- MacBook Pro Early 2011
  - Intel® Core i5-2415M
  - SandyBridge

- Dell Latitude E7470
  - Intel® Core i5-6300U
  - SkyLake

- Compiler:
  - clang compiler version 6.1.0 (clang-602.0.53)
  - gcc compiler version 5.4.0 (5.4.0-6ubuntu1-16.04.2)

- Compiler flags: `-Ofast -fno-stack-protector -march=native`

- Operating System in Single User mode to get rid of noise (e.g. context switches)

- Calculate the median of 91 averaged timings of 200 measurements [KR11]
Table: Real-world use case settings for our benchmarking.

<table>
<thead>
<tr>
<th>Message Size</th>
<th>Associated Data Size</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>5 byte</td>
<td>one keystroke (e.g. SSH)</td>
</tr>
<tr>
<td>16 bytes</td>
<td>5 byte</td>
<td>small payload</td>
</tr>
<tr>
<td>557 byte</td>
<td>5 byte</td>
<td>average IP packet size$^3$</td>
</tr>
<tr>
<td>1.5 kB</td>
<td>5 byte</td>
<td>ethernet MTU, TLS</td>
</tr>
<tr>
<td>16 kB</td>
<td>5 byte</td>
<td>max TCP packet size</td>
</tr>
<tr>
<td>1 MB</td>
<td>5 byte</td>
<td>file upload</td>
</tr>
</tbody>
</table>

$^2$[http://netsekure.org/2010/03/tls-overhead](http://netsekure.org/2010/03/tls-overhead)

Results

1. Classification of the 2\textsuperscript{nd} round CAESAR Candidates

2. Software Optimizations

3. Benchmarking Framework

4. Results

5. Conclusions
Comparison of all CAESAR 2\textsuperscript{nd} Round Candidates

![Graph showing performance (cpb) vs. message length (bytes) for various CAESAR candidates.]

- acorn128v2_opt
- aesdaaes256ocbtaglen128v1_opt
- aegis128l_aesnib
- aes128gcmv1_openssl
- aes128n8t8ilocv2_aesni
- aes128n8t8silcv2_aesni
- aes128otrpv3_nip7m1
- aescopav2_ref
- aesjambuv2_aesni
- aezv4_aesni
- acon128avv11_opt64
- deoxyseq256128v13_aesni
- elmd600v2_ref
- hsiivlov1_ref
- icepole128av2_ref
- joltikseq6464v13_ref
- ketjersrv1_reference
- minalpherv11_ref
- morus128256v1_avx2
- norx6441_ymm
- omdsha512k512n256tau256v2_avx1
- paeq64_aesni
- pit4cipher256v2_goptv
- poetv2aes4_ni
- primatesv1gibbon80_ref
- scream10v3_sse
- seakeyakv2_SandyBridge
- shellaes128v2d4n80_ref
- stribob192v2_sse3
- liaoaxinv2_nim
- trivia0v2_sse4

Ralph Ankele - Royal Holloway, University of London
Software Benchmarking of the 2\textsuperscript{nd} round CAESAR Candidates
Comparison of all Block Cipher based schemes

Ralph Ankele - Royal Holloway, University of London
Software Benchmarking of the 2\textsuperscript{nd} round CAESAR Candidates
Comparison of all Sponge based schemes

Message length (bytes)

Performance (cpb)

- aes128gcmv1 OpenSSL
- ascon128av1_opt64
- icepole128av2_ref
- ketjersv1_reference
- norx6441_ymm
- pi64cipher256v2_goptv
- primatesv1gibbon80_ref
- seakeyakv2_SandyBridge
- stribob192r2_sse3

Ralph Ankele - Royal Holloway, University of London Software Benchmarking of the 2nd round CAESAR Candidates slide 27 /39
Comparison of all Stream Cipher based schemes

<table>
<thead>
<tr>
<th>Message length (bytes)</th>
<th>Performance (cpb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.93</td>
</tr>
<tr>
<td>128</td>
<td>25.16</td>
</tr>
<tr>
<td>256</td>
<td>10.48</td>
</tr>
<tr>
<td>384</td>
<td>8.25</td>
</tr>
<tr>
<td>512</td>
<td>7.07</td>
</tr>
<tr>
<td>640</td>
<td>1.65</td>
</tr>
<tr>
<td>768</td>
<td>2.53</td>
</tr>
<tr>
<td>896</td>
<td>1.35</td>
</tr>
<tr>
<td>1024</td>
<td>1.08</td>
</tr>
<tr>
<td>1152</td>
<td>0.93</td>
</tr>
<tr>
<td>1280</td>
<td>0.85</td>
</tr>
<tr>
<td>1408</td>
<td>0.79</td>
</tr>
<tr>
<td>1536</td>
<td>0.73</td>
</tr>
<tr>
<td>1664</td>
<td>0.67</td>
</tr>
<tr>
<td>1792</td>
<td>0.61</td>
</tr>
<tr>
<td>1920</td>
<td>0.55</td>
</tr>
<tr>
<td>2048</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Legend:
- acorn128v2_opt
- aes128gcmv1_openssl
- hs1sivlov1_ref
- morus1280256v1_avx2
- trivia0v2_sse4
Comparison of all Permutation based schemes

<table>
<thead>
<tr>
<th>Message length (bytes)</th>
<th>Performance (cpb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.51</td>
</tr>
<tr>
<td>128</td>
<td>2.53</td>
</tr>
<tr>
<td>256</td>
<td>5.09</td>
</tr>
<tr>
<td>512</td>
<td>11.22</td>
</tr>
<tr>
<td>1024</td>
<td>609.55</td>
</tr>
</tbody>
</table>

- aes128gcmv1_openssl
- minalpherv11_ref
- paeq64_aesni

Ralph Ankele - Royal Holloway, University of London
Software Benchmarking of the 2\textsuperscript{nd} round CAESAR Candidates
Comparison of all Compression Function based schemes
Comparison in the TLS setting

- joltieqv1284v13_ref
- primesv1gibbon80_ref
- minalpherv11_ref
- aescopav2_ref
- shellaes128v2d8n80_ref
- ketjesrv1_reference
- stribob192v2_sse3
- omdsha512x128n128tau128v2_sse4
- icepole128av2_ref
- scream10v3_sse
- acorn128v2_opt
- trivialv2_sse4
- pi64cipher128v2_goptv
- hs1sivov1_ref
- aesjamuv2_aesni
- asconv128av11_opt64
- paeq80_aesni
- lakekeyakv2_generic64
- aes128n8t8silicv2_aesni
- aes128n12t8clocv2_aesni
- norx6441_ym
- **aesc128cmv1_openssl**
- poetv2aes4_ni
- deoxyseaq128128v13_aesni
- aeadav128ocbtglen128v1_opt
- morus1280256v1_avx2
- aes128otrpv3菥p7m2
- aezv4_aesni
- aegis128i_aesnic
- tiaoxinv2_nim

Performance (cpb)

![Graph showing comparison of different algorithms in the TLS setting](image-url)
Comparison in the SSH setting

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Performance (cpb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>primatesv1gibbon80_ref</td>
<td>2467.21</td>
</tr>
<tr>
<td>joltilikeq12864v13_ref</td>
<td>6969.07</td>
</tr>
<tr>
<td>minalpherv11_ref</td>
<td>3479.76</td>
</tr>
<tr>
<td>shellaes128v2d8n80_ref</td>
<td>1267.94</td>
</tr>
<tr>
<td>omdsha512k128n128tau128v2_sse4</td>
<td>3192.2</td>
</tr>
<tr>
<td>aescopav2_ref</td>
<td>3954.52</td>
</tr>
<tr>
<td>scream10v3_sse</td>
<td>712.97</td>
</tr>
<tr>
<td>icepole128av2_ref</td>
<td>3466.63</td>
</tr>
<tr>
<td>pi64cipher128v2_goptv</td>
<td>3459.96</td>
</tr>
<tr>
<td>keltexav1_reference</td>
<td>3475.54</td>
</tr>
<tr>
<td>stribob192v2_sse9</td>
<td>4401.03</td>
</tr>
<tr>
<td>acom128v2_opt</td>
<td>369.85</td>
</tr>
<tr>
<td>trivia0v2_sse4</td>
<td>267.01</td>
</tr>
<tr>
<td>hs1silv1v1_ref</td>
<td>245.93</td>
</tr>
<tr>
<td>lakekeyalv2_generic64</td>
<td>205.17</td>
</tr>
<tr>
<td>aes128gcmv1_openssl</td>
<td>219.41</td>
</tr>
<tr>
<td>aeadaes128ocbtaglen128v1_opt</td>
<td>195.06</td>
</tr>
<tr>
<td>norx6441_ymmm</td>
<td>150.96</td>
</tr>
<tr>
<td>paeq80_aesni</td>
<td>144.74</td>
</tr>
<tr>
<td>poetv2aes4_i3</td>
<td>130.00</td>
</tr>
<tr>
<td>deoxysneq128128v13_aesni</td>
<td>101.41</td>
</tr>
<tr>
<td>morus1280256v1_avx2</td>
<td>196.04</td>
</tr>
<tr>
<td>aesc128n88silicv2_aesni</td>
<td>803.03</td>
</tr>
<tr>
<td>ascon128av11_opt64</td>
<td>714.45</td>
</tr>
<tr>
<td>aesc128n128cloccv2_aesni</td>
<td>67.45</td>
</tr>
<tr>
<td>aessjambuv2_aesni</td>
<td>65.53</td>
</tr>
<tr>
<td>aesc128otp4v3_mp7m2</td>
<td>56.47</td>
</tr>
<tr>
<td>tiaoxinv2_nim</td>
<td>55.66</td>
</tr>
<tr>
<td>aezv4_aesni</td>
<td>50.44</td>
</tr>
<tr>
<td>aegis128l_aesnic</td>
<td>43.42</td>
</tr>
</tbody>
</table>

Ralph Ankele - Royal Holloway, University of London
Software Benchmarking of the 2nd round CAESAR Candidates
Currently fastest cipher (Software)

Figure: Tiaoxin v2.0 (SSE and AES-NI optimized)
Conclusions

1. Classification of the 2nd round CAESAR Candidates

2. Software Optimizations

3. Benchmarking Framework

4. Results

5. Conclusions
Conclusions

- New framework to benchmark Authenticated Encryption ciphers
  - Very simple, only focus on AE ciphers
  - Timer Stamp Counter (with optimized RDTSCP instruction)
  - Reduction of noise during measurements
- Comparison of CAESAR 2nd round Candidates
  - TLS setting
  - SSH setting
- 23 out of 30 ciphers offer at least one optimization
Further Work

OPTIMIZE

ALL THE CIPHERS!!!
Thank you for your attention!
Daniel J. Bernstein. 
Supercop. 

Shay Gueron. 
Intel® advanced encryption standard (aes) new instructions set. 

Ted Krovetz and Phillip Rogaway. 
The Software Performance of Authenticated-Encryption Modes, 
pages 306–327. 

Gregory Lento. 
Optimizing performance with intel® advanced vector extensions. 
performance-xeon-e5-v3-advanced-vector-extensions-paper. 
html, 2014.

Neon.

Markku-Juhani Saarinen. The BRUTUS automatic cryptanalytic framework.