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 \quad \textsuperscript{2}University of Oxford, UK

October 20, 2016

SPEED-B, Utrecht, The Netherlands
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

---

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 competition

CAESAR

- Competition for Authenticated Encryption: Security, Applicability and Robustness
- 57 first round candidates (9 withdrawn)
- 30 second round candidates
- 16 third round candidates

Application of AE

- IPsec, SSL/TLS, SSH
CAESAR competition

CAESAR

- **Competition for Authenticated Encryption:** Security, Applicability and Robustness
- 57 first round candidates (9 withdrawn)
- 30 second round candidates
- 16 third round candidates

Application of AE

- IPsec, SSL/TLS, SSH
CAESAR competition

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
Underlying Primitive

- AES: 10
- Others: 3
- AES Round: 9
- SPN: 3
- Keccak: 3
- Dedicated Permutation: 1
- Dedicated Stream Cipher: 1
- Dedicated Block Cipher: 1
- SHA2: 1
- LRX: 1
- ARX: 1
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
Security Proof

Yes: 24
No: 6

Ralph Ankele - Royal Holloway, University of London
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
Software Optimizations

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

2. Software Optimizations

3. Benchmarking Framework

4. Results

5. Conclusions
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. AESENCE, 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]
NEON

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
NEON

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 2\textsuperscript{nd} 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
- \texttt{time()} and \texttt{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]
- Uses timer stamp counter as timer (with RDTSC and CPUID)
- Recommends turn-off of hyper threading/idle state during measurements
- Complex benchmarking framework for cryptographic primitives

BRUTUS [Saa16]
- Uses clock() as timer
- No noise reduction
- Small codebase, rapid testing cycle

Our Framework
- Optimized timer stamp counter (i.e. RDTSCP) [Pao10]
- Reduction of noise using single-user mode, averaging and median
- Focus on authenticated encryption and real-world usecases
Benchmarking Framework

SUPERCOP [Ber16]
- Uses timer stamp counter as timer (with RDTSC and CPUID)
- Recommends turn-off of hyper threading/idle state during measurements
- Complex benchmarking framework for cryptographic primitives

BRUTUS [Saa16]
- Uses clock() as timer
- No noise reduction
- Small codebase, rapid testing cycle

Our Framework
- Optimized timer stamp counter (i.e. RDTSCP) [Pao10]
- Reduction of noise using single-user mode, averaging and median
- Focus on authenticated encryption and real-world usecases
Benchmarking Framework

SUPERCOP [Ber16]

- Uses timer stamp counter as timer (with RDTSC and CPUID)
- Recommends turn-off of hyper threading/idle state during measurements
- Complex benchmarking framework for cryptographic primitives

BRUTUS [Saa16]

- Uses clock() as timer
- No noise reduction
- Small codebase, rapid testing cycle

Our Framework

- Optimized timer stamp counter (i.e. RDTSCP) [Pao10]
- Reduction of noise using single-user mode, averaging and median
- Focus on authenticated encryption and real-world usecases
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]
## Benchmarking Settings and Real-World Usecases

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&lt;sup&gt;2&lt;/sup&gt;</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>

<sup>2</sup> [http://netsekure.org/2010/03/tls-overhead](http://netsekure.org/2010/03/tls-overhead)

Results

1. Classification of the 2nd round CAESAR Candidates

2. Software Optimizations

3. Benchmarking Framework

4. Results

5. Conclusions
Comparison of all CAESAR 2\textsuperscript{nd} Round Candidates
Comparison of all Block Cipher based schemes

Message length (bytes)

Performance (cpb)

Ralph Ankele - Royal Holloway, University of London

Software Benchmarking of the 2nd round CAESAR Candidates
Comparison of all Sponge based schemes

![Graph showing performance vs message length for various Sponge-based schemes]
Comparison of all Stream Cipher based schemes

![Graph comparing performance of different Stream Cipher schemes](image-url)

- acorn128v2_opt
- aes128gcmv1_openssl
- hs1sivlov1_ref
- morus1280256v1_avx2
- trivia0v2_sse4

*Message length (bytes)*

*Performance (cpb)*
Comparison of all Permutation based schemes

![Comparison of all Permutation based schemes](image)

Ralph Ankele - Royal Holloway, University of London

Software Benchmarking of the 2nd round CAESAR Candidates
Comparison of all Compression Function based schemes

![Graph showing performance vs message length for compression functions.](image-url)
Comparison in the TLS setting

Performance (cpb)

10^0 10^1 10^2 10^3

joliteq12864v13_ref
primatesv1gibbon80_ref
minalpherv11_ref
aescopav2_ref
shellaes128v2d8n80_ref
ketjesrv1_reference
stribob1922_sse3
omdsha512k128n128tau128v2_sse4
icepole128av2_ref
scream10v3_sse
acorn128v2_opt
trivialv2_sse4
pi64cipher128v2_goipv
hs1sivov1_ref
aesjambuv2_aesni
acon128av11_opt64
paepq0_aesni
lakekeyakv2_generic64
aes128n8t8silicv2_aesni
aes128n12tclocv2_aesni
norx6441_ymm
aes128cmv1_openssl
poetv2aes4_ni
deoxysneq128128v13_aesni
aedaes128octaglen128v1_opt
morus128o256v1_avx2
aesc128otrpv3_nip7m2
aexv4_aesni
eaegi128l_aesnic
tiaoxinv2_nim

Performance (cpb)
Comparison in the SSH setting

- aegis128l_aesnic
- aezv4_aesni
- tiaoxinv2_nim
- aes128otrpv3_nip7m2
- aesjambuv2_aesni
- a...
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 2\textsuperscript{nd} round Candidates
  - TLS setting
  - SSH setting
- 23 out of 30 ciphers offer at least one optimization
Further Work

OPTIMIZE

ALL THE CIPHERS!!!
Questions?

Thank you for your attention!
References I

Supercop. 

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

The Software Performance of Authenticated-Encryption Modes, 
pages 306–327. 

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