AES-Based Authenticated Encryption Modes in Parallel High-Performance Software

Andrey Bogdanov Martin M. Lauridsen Elmar Tischhauser

mmeh @ dtu.dk

DTU Compute, Technical University of Denmark

DIAC 2014
Santa Barbara, August 24, 2014

Context

Context

- Huge interest in AE in symmetric community due to **CAESAR**
- Focus on AEAD **modes of operation** for block ciphers
- Block cipher: **AES-128**
- Intel's latest **Haswell** architecture (2013) improves AEAD-relevant instructions
  - **AES-NI** instructions
  - `pclmulqdq`: Used for multiplication in $GF(2^n)$
- Machine: Intel(R) Core(TM) i5-4300U CPU @ 1900 MHz

Nonce-based vs. nonce-free

In this talk...

Nonce-based modes

- Lose authenticity, privacy or both when the nonce requirement is violated

Nonce-free modes

- Maintain authenticity and privacy up to the common message prefix

AEAD modes covered

Modes implemented in this work

	First AES-NI	First Haswell AES-NI
Nonce-based	**OTR** **CLOC** COBRA (FSE 2014) **SILC**	CCM **OCB3**
Nonce-free	McOE-G **POET** (hash: AES-128) **Julius** (Julius-ECB)	**COPA**

Also implemented: **JAMBU** and GCM.

(**CAESAR** submissions **in bold**)

Multiple-message setting

Multiple-message setting I

Internet packet sizes essentially follow a **bimodal distribution**

- 44% of packets: 40-100 bytes
- 37% of packets: 1400-1500 bytes

Thus, the CAESAR portfolio

- Should have **excellent performance for messages up to 2KB**
- This is the range we benchmark in this work

Wolfgang John and Sven Tafvelin

Analysis of Internet backbone traffic and header anomalies observed

In Internet Measurement Conference 2007, pages 111–116.

David Murray and Terry Koziniec

The state of enterprise network traffic in 2012

In 18th Asia-Pacific Conference on Communications 2012

Multiple-message setting II

Meanwhile, this poses a problem

1) Most AEAD modes obtain their best performance **only for long messages**

Another, mostly unrelated problem

2) Sequential AEAD modes **can not fully utilize pipeline** for AES encryption on general-purpose CPUs

To remedy these two problems

- We consider processing **multiple independent message streams** in parallel as **part of the algorithm itself**
- Using **varying parallelism degrees** for all twelve AEAD modes
- We are **not suggesting** to implement message scheduling!

Introduced with the performance study of ALE from FSE 2013

Example: AES-CBC in a perfect world I

- In a **perfect world**, all messages have equal length!

# msgs.	cycles/byte	speed-up
single msg.	4.28	–
2	**2.15**	**×1.99**
3	**1.43**	**×2.99**
4	**1.08**	**×3.96**
5	0.88	×4.86
6	0.74	×5.78
7	0.64	×6.69
8	0.63	×6.79

- Speed-up nearly linear for **2 through 4 multiple messages**

Example: AES-CBC in a perfect world II

Does parallel messages imply increased latency?

- For perfect parallelization, **no increase in latency**

Latencies for processing

- Single message: $4.28 \cdot |M|$ cycles
- 2 parallel messages: $4.30 \cdot |M|$ cycles
- 3 parallel messages: $4.29 \cdot |M|$ cycles
- 4 parallel messages: $4.32 \cdot |M|$ cycles

With 8 parallel messages

- Latency increased by 18%
- Throughput increased $\times 6.8$

Example: AES-CBC in a realistic world

Assume we process 4 messages in parallel

- 2 messages of 128 bytes
- 1 message of 512 bytes
- 1 message of 1024 bytes

Actual speedup

$$= \frac{\text{cycles in single-message setting}}{\text{cycles in multiple-message setting}}$$

$$= \frac{4.28 \cdot (2 \cdot 128 + 512 + 1024) \text{ cycles}}{1.09 \cdot 4 \cdot 128 + 2.15 \cdot 2 \cdot (512 - 128) + 4.28 \cdot (1024 - 512) \text{ cycles}}$$

$$= \mathbf{1.74}$$

- Factor 2.27 slowdown from perfect world to realistic world

Performance data

Performance data: Baseline

Mode	Single msg.	Multiple msg. (# msgs.)
AES-ECB	0.63	0.63 (8)
AES-CTR	0.74	0.75 (8)
AES-CBC	4.28	0.63 (8)

Theoretical minimum of $\approx 10/16 = 0.625$ cpb obtained for AES-ECB

AES-CBC obtains the same with 8 parallel messages (in a perfect world)

Performance data: Single-message setting

Mode	Message length (bytes)				
	128	256	512	1024	2048
	single message				
CCM	5.35	5.19	5.14	5.11	5.10
GCM	2.09	1.61	1.34	1.20	1.14
OCB3	2.19	1.43	1.06	0.87	0.81
OTR	2.97	1.34	1.13	1.02	0.96
CLOC	4.50	4.46	4.44	4.46	4.44
COBRA	4.41	3.21	2.96	2.83	2.77
JAMBU	9.33	9.09	8.97	8.94	8.88
SILC	4.57	4.54	4.52	4.51	4.50
McOE-G	7.77	7.36	7.17	7.07	7.02
COPA	3.37	2.64	2.27	2.08	1.88
POET	5.30	4.93	4.75	4.68	4.62
Julius	4.18	4.69	3.24	3.08	3.03

Performance data: Multiple-message setting

Mode		Message length (bytes)				
		128	256	512	1024	2048
	# msgs.	multiple messages				
CCM	8	1.51	1.44	1.40	1.38	1.37
GCM	13	1.81	1.72	1.68	1.65	1.64
OCB3	7	1.59	1.16	0.94	0.83	0.77
OTR	8	1.28	1.08	0.98	0.94	0.92
CLOC	7	1.40	1.31	1.26	1.24	1.23
COBRA	8	2.04	1.88	1.80	1.76	1.75
JAMBU	14	2.14	1.98	1.89	1.85	1.82
SILC	7	1.43	1.33	1.28	1.25	1.24
McOE-G	7	1.91	1.76	1.68	1.64	1.62
COPA	15	1.62	1.53	1.48	1.46	1.45
POET	8	3.24	2.98	2.86	2.79	2.75
Julius	7	2.53	2.27	2.16	2.09	2.06

Performance data: Speed-ups

	Message length (bytes)				
Mode	128	256	512	1024	2048
CCM	×3.54	×3.60	×3.67	×3.70	×3.72
GCM	×1.15	×0.94	×0.80	×0.73	×0.70
OCB3	×1.38	×1.23	×1.13	×1.05	×1.05
OTR	×2.32	×1.24	×1.15	×1.09	×1.04
CLOC	×3.21	×3.40	×3.52	×3.60	×3.61
COBRA	×2.16	×1.71	×1.64	×1.61	×1.58
JAMBU	×4.36	×4.59	×4.75	×4.83	×4.88
SILC	×3.20	×3.41	×3.53	×3.61	×3.63
McOE-G	×4.07	×4.18	×4.27	×4.31	×4.33
COPA	×2.08	×1.73	×1.53	×1.42	×1.30
POET	×1.64	×1.65	×1.66	×1.68	×1.45
Julius	×1.65	×2.07	×1.50	×1.47	×1.47

Another example: SILC in the multiple-message setting

In a perfect world

- Speed-up roughly $\times 3.60$ using 7 multiple messages

In a realistic world

- Assume we process 7 messages in parallel
  - 4 messages of 128 bytes
  - 3 messages of 2048 bytes

$$\begin{aligned}\text{Actual speedup} &= \frac{\text{cycles in single-message setting}}{\text{cycles in multiple-message setting}} \\ &= \frac{4.57 \cdot 4 \cdot 128 + 4.50 \cdot 3 \cdot 2048 \text{ cycles}}{1.24 \cdot 7 \cdot 128 + 1.76 \cdot 3 \cdot (2048 - 128) \text{ cycles}} \\ &= \mathbf{2.67}\end{aligned}$$

- Factor 1.35 slowdown from perfect world to realistic world

Summary

- AEAD modes should excel for messages up to 2KB
- Obtained first AES-NI and/or Haswell performance figures for many new (CAESAR candidate) AEAD modes
- Multiple-message processing allows significant speed-up of especially sequential modes
  - Also for messages of varying length

Read the full version of the paper at

`https://eprint.iacr.org/2014/186`

(also has nice pictures)

Thanks.