Directions in Authenticated Ciphers
DIAC 2013, 11–13 August 2013, Chicago, USA

AES-GCM software performance on the current high end CPUs as a performance baseline for CAESAR competition

Shay Gueron

University of Haifa
Department of Mathematics, Faculty of Natural Sciences, University of Haifa, Israel

Intel Corporation
Israel Development Center, Haifa, Israel

shay@math.haifa.ac.il shay.gueron@intel.com

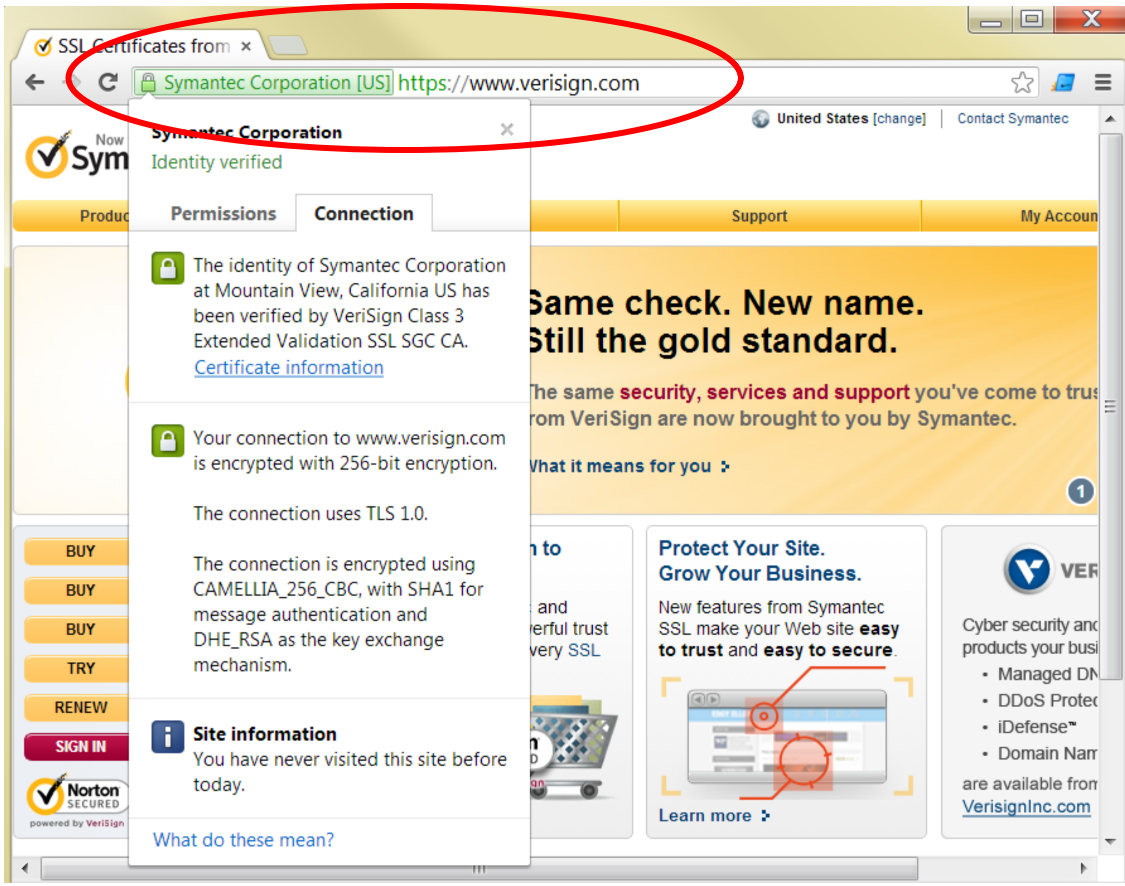
Agenda

- We will explore the impact of an efficient Authenticated Encryption (AES-GCM) at the system level
- Set a performance bar for the CAESAR competition

Background and motivation

Background

- We live in an https:// world
  - TLS/SSL protected client-server communications
- Why not https everywhere?
- Due to **overheads' costs**
  - Cryptographic algorithms for secure communications ➔ computational overhead



Symantec Corporation

Identity verified

Permissions | Connection

The identity of Symantec Corporation at Mountain View, California US has been verified by VeriSign Class 3 Extended Validation SSL SGC CA.

Certificate information

Your connection to www.verisign.com is encrypted with 256-bit encryption.

The connection uses TLS 1.0.

The connection is encrypted using CAMELLIA_256_CBC, with SHA1 for message authentication and DHE_RSA as the key exchange mechanism.

Site information

You have never visited this site before today.

What do these mean?

- SSL/TLS secure webservers are widely used, e.g., for
  - Banking
  - Ecommerce (eBay, Amazon, PayPal®…)
  - Google searches
  - Social networks (e.g., Facebook)
  - File hosting (Dropbox, Skydrive, Google Drive)

client – server handshake

public key based; a fixed (large) computational payload

Client Server

Traffic

Application Data: Authenticated Encryption

Popular cipher choices used today

- ~~RC4 + HMAC-MD5~~
- RC4 + HMAC-SHA-1
- AES + HMAC-SHA-1

AES-GCM is a more efficient Authenticated Encryption scheme

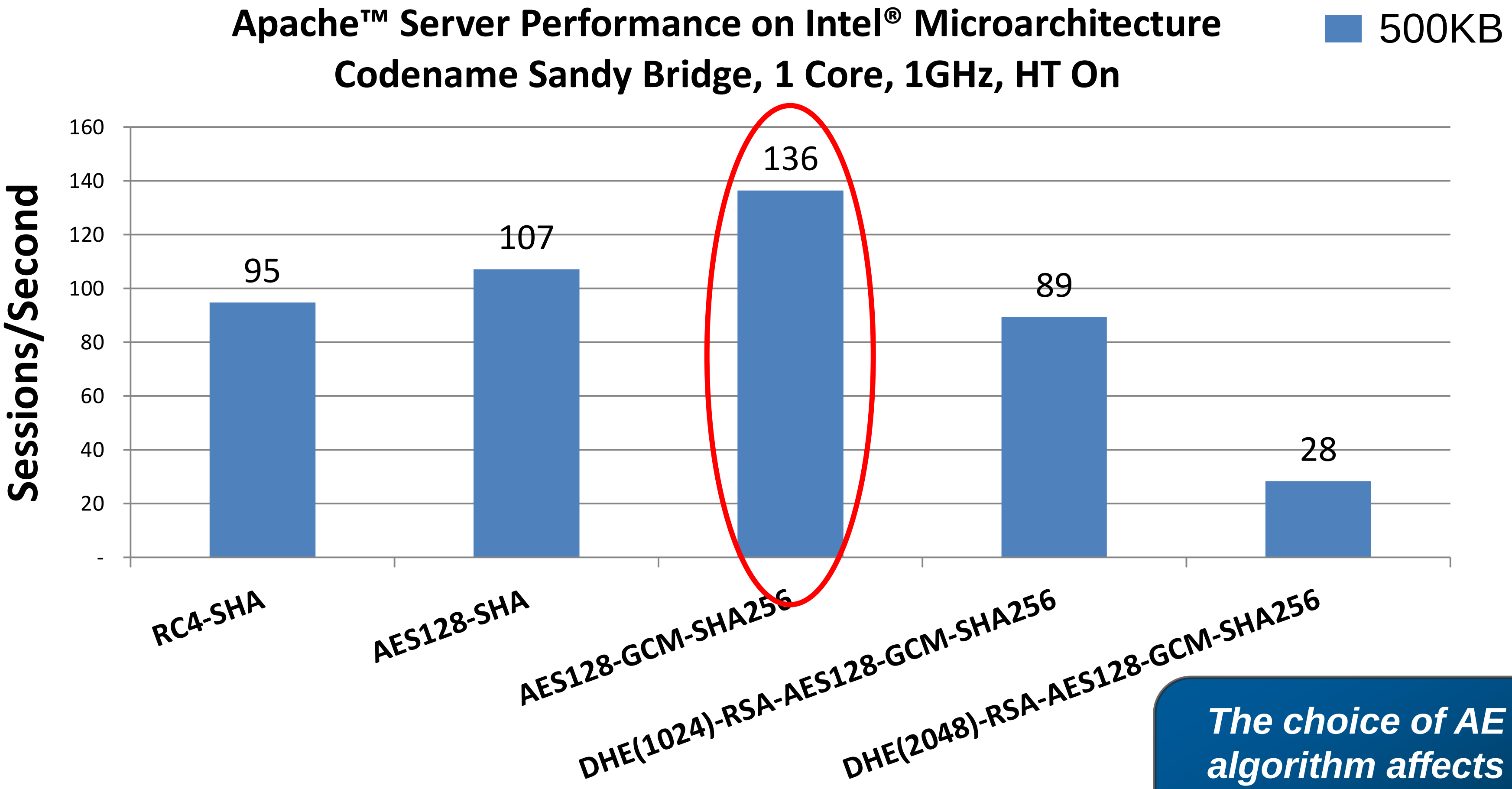
The effect of the choice of AE on the servers' efficiency

Different Types of Workloads

- Main two steps of the SSL connection
  - Establish a shared secret through an SSL handshake
  - Use the shared secret to derive symmetric keys to authenticate/encrypt all subsequent traffic
- What dominates the workload?
  - Handshake dominates when the files are small
  - Authenticated Encryption dominates when the files are large
- File sizes depends on the type of the application;
- Here are some example
  - Webmail login to check the last mail (textual): 470 KB
  - Login to a bank account and check balance: 320 KB
  - Facebook login and scroll through 5 pictures a friend posted: 850KB
  - File hosting services: typically use large files (up to a few GB)

Client-server communications are affected differently by AE performance

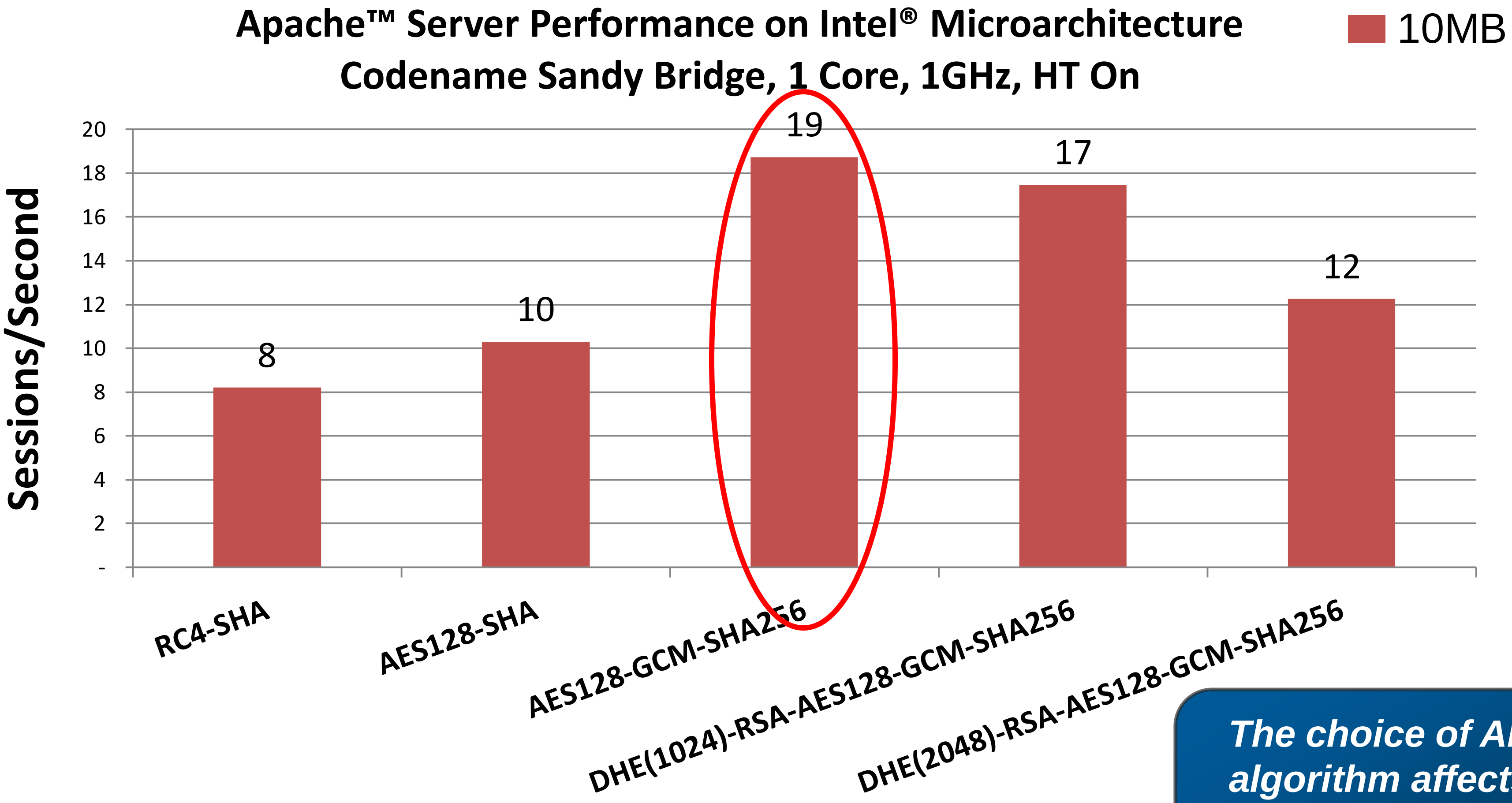
SSL Server Performance small data sessions (500KB)



The choice of AE algorithm affects directly servers' efficiency

*Tested Apache™ server uses OpenSSL® development version

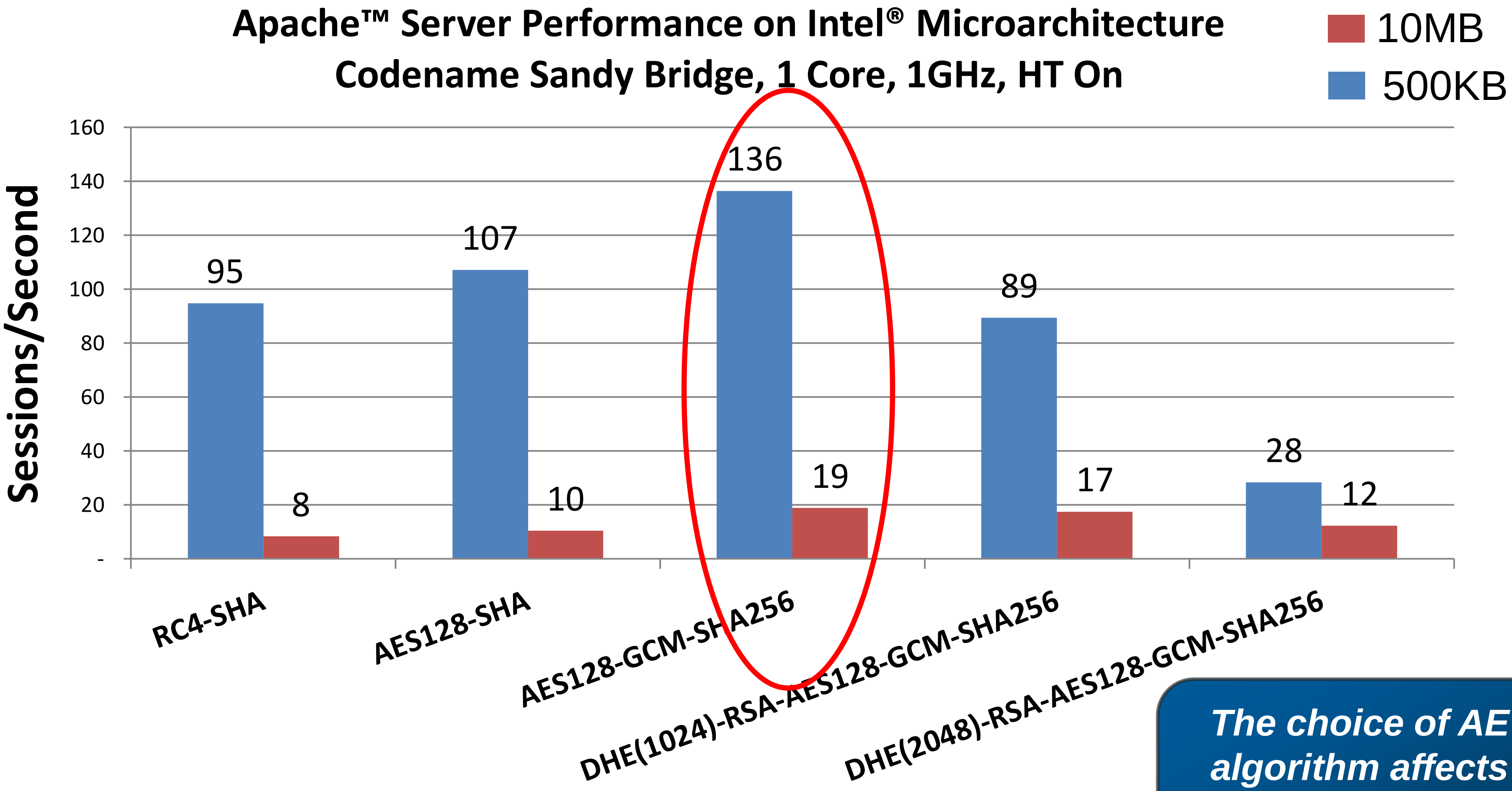
SSL Server Performance large-data sessions (10MB)



The choice of AE algorithm affects directly servers' efficiency

*Tested Apache™ server uses OpenSSL® development version

SSL Server Performance …small and large…



*Tested Apache™ server uses OpenSSL® development version

The choice of AE algorithm affects directly servers' efficiency

AES software optimization and the latest open source patches

AES-GCM and Intel’s AES-NI / PCLMULQDQ

- Intel introduced a new set of instructions (2010)
- AES-NI:
  - Facilitate high performance AES encryption and decryption
- PCLMULQDQ **64 x 64 → 128 (carry-less)**
  - Binary polynomial multiplication; speeds up computations in binary fields
- Has several usages --- AES-GCM is one
- To use it for the GHASH computations: GF(2^{128}) multiplication:
  1. Compute **128 x 128 → 256** via carry-less multiplication (of 64-bit operands)
  2. Reduction: **256 → 128 *“modulo“*** **$x^{128} + x^7 + x^2 + x + 1$** (done efficiently via software)

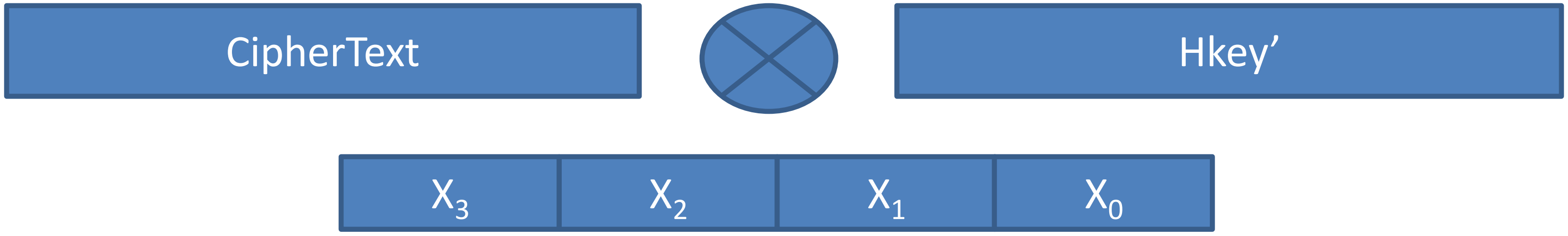
AES-NI and PCLMULQDQ can be used
for speeding up AES-GCM Authenticated Encryption

Recent open source contributions The new AES-GCM related patches

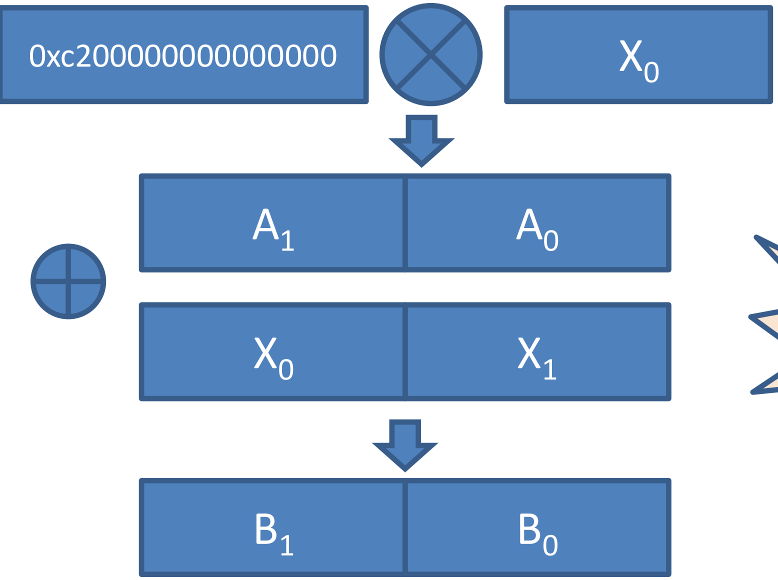
- Sept./Oct. 2012: We published two patches for two popular open source distributions: OpenSSL® and NSS
  - Authors: S. Gueron and V. Krasnov
  - Inherently side channel protected
    - “constant time” in the strict definition
  - Fast on the current x86-64 processors (Intel® Microarchitectures Codename Sandy Bridge, Ivy Bridge and Haswell)
    - The (we can do) best balance between slower CLMUL implementation of Sandy Bridge and Ivy Bridge Microarchitectures, and the faster CLMUL of the Haswell Microarchitecture
  - Status: integrated into NSS, the ideas adopted and integrated into OpenSSL
- March 2013: We published a patch for OpenSSL accelerating AES-CTR performance
  - AES-CTR is the encryption mode, underlying the AES-GCM
  - Status: mostly adopted by OpenSSL
- May 2013: We published a patch for NSS, performing side-channel protected GHASH, for generic processors
- Let’s review how this was done

AES-GCM software optimization highlights

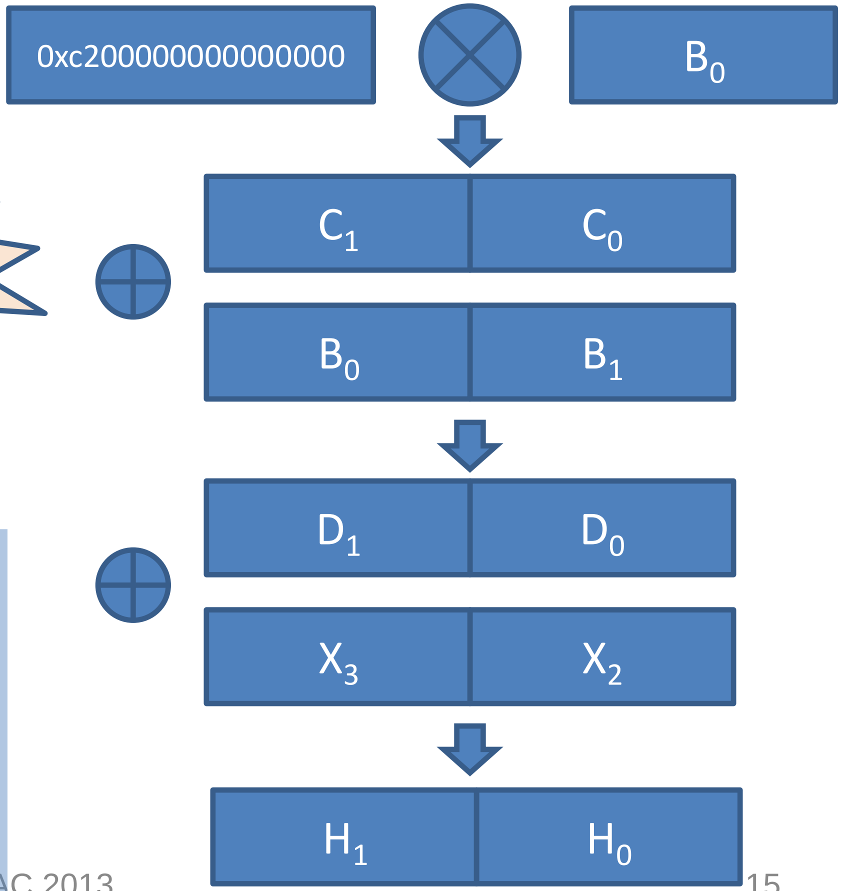
- Carry-less Karatsuba multiplication
  - Best on Sandy Bridge / Ivy Bridge Microarchitectures (slower PCLMULQDQ)
- Schoolbook method for Haswell Microarchitecture
  - Haswell has improved PCLMULQDQ
- New reduction algorithm
  - Carry-less Montgomery for the GHASH operations [Gueron 2012]
- Encrypt 8 counter blocks
- Deferred reduction (using 8 block aggregation)
- Fixed elements outside the brackets
- Interleave CTR and GHASH
  - Gains 3% over encrypting and MAC-ing serially
- Inherently side channel protected
  - “constant time” in the strict definition
- The challenge: delicate balance on the Sandy Bridge/Ivy Bridge/Haswell Microarchitectures



The optimized reduction [Gueron]



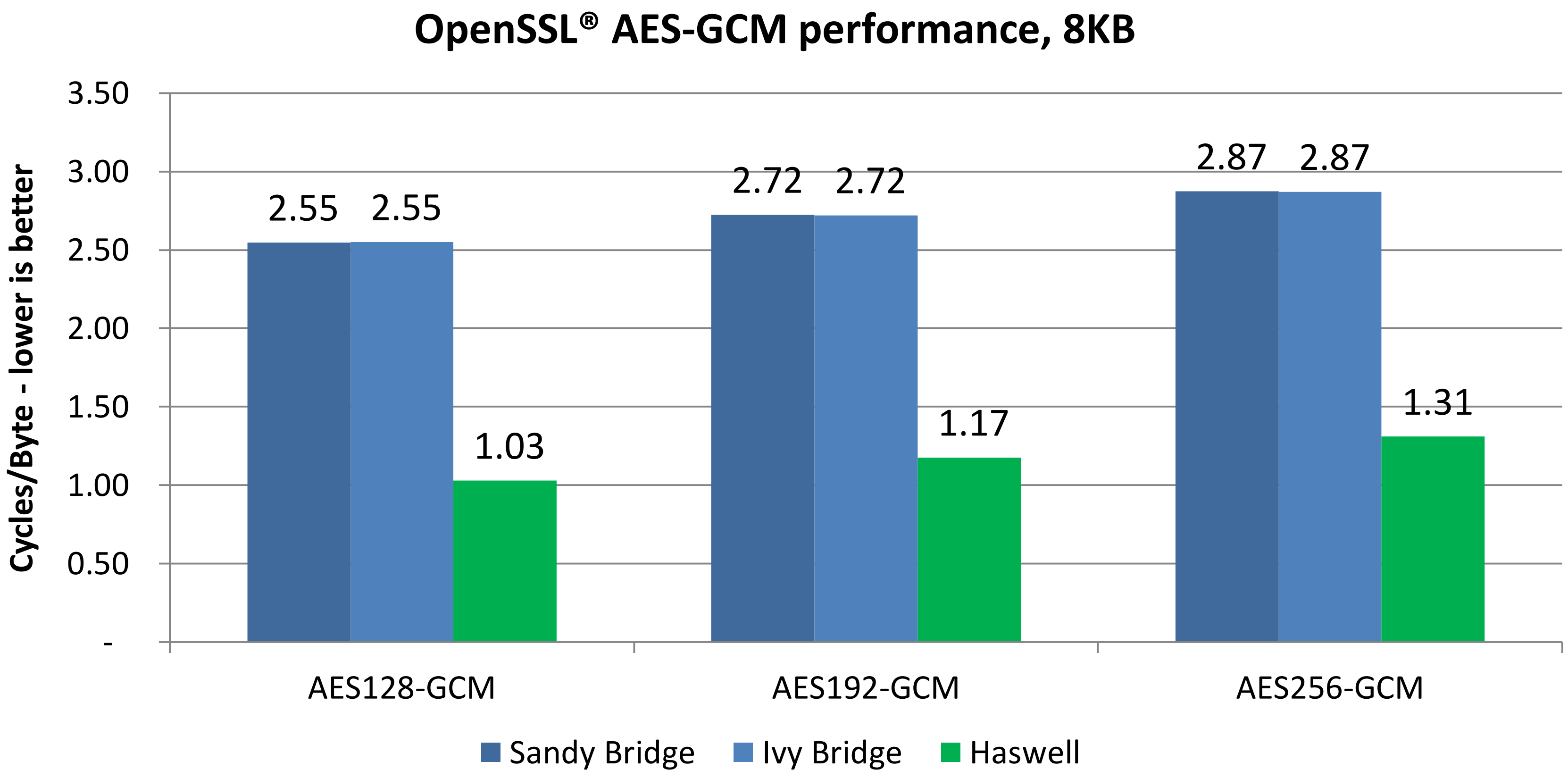
That’s all



```
vmovdqa    T3, [W]
vpclmulqdq T2, T3, T7, 0x01
vpshufd    T4, T7, 78
vpxor      T4, T4, T2
vpclmulqdq T2, T3, T4, 0x01
vpshufd    T4, T4, 78
vpxor      T4, T4, T2
vpxor      T1, T1, T4   ; result in T1
```

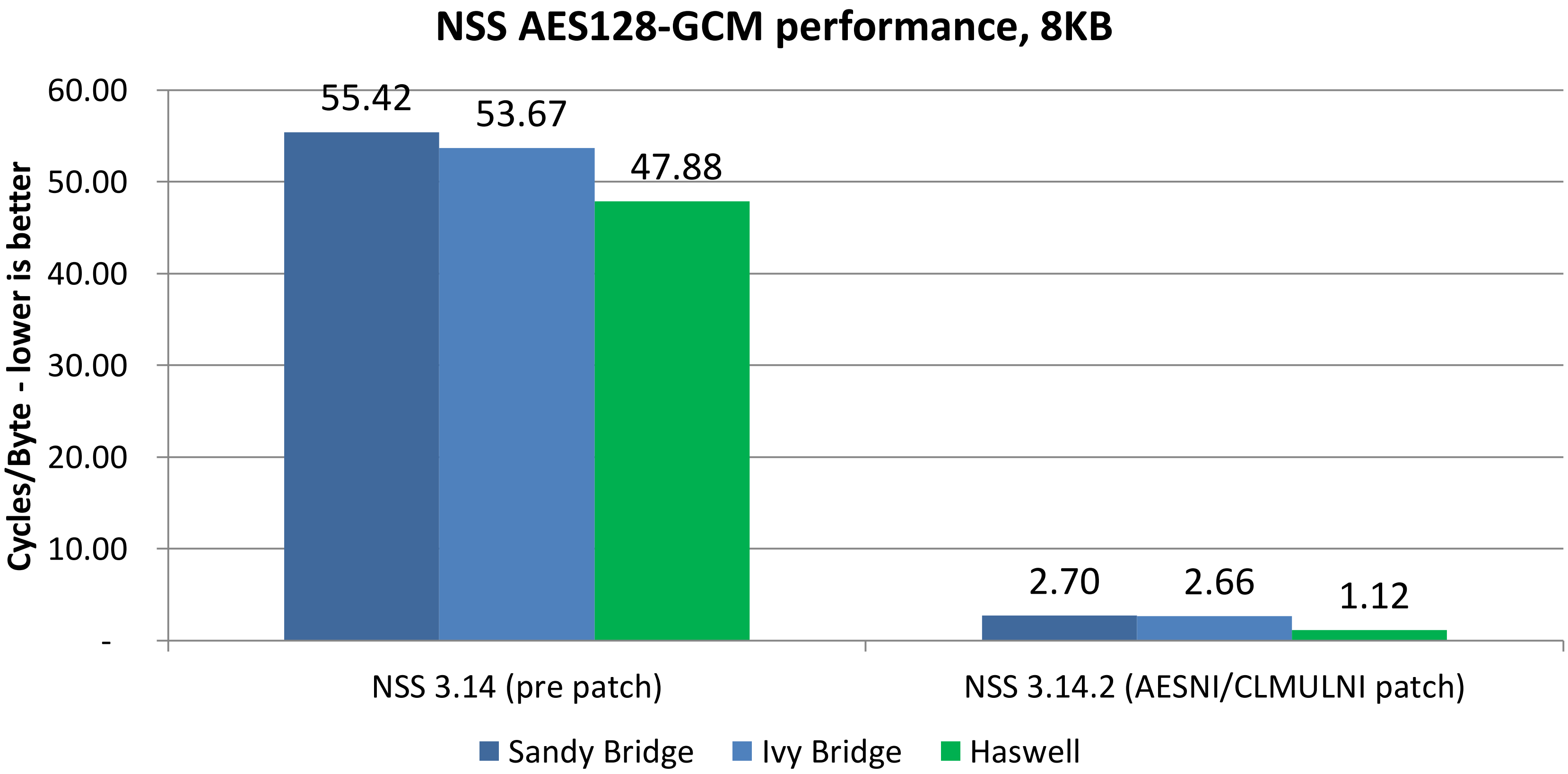
Performance numbers

AES-GCM performance on OpenSSL® (development version)



A significant (more than 2x) performance improvement on the new Intel® Microarchitecture Codename Haswell

AES128-GCM performance on NSS 3.14.2



A significant performance improvement on the new version of NSS
More than 2x speedup on the new
Intel® Microarchitecture Codename Haswell

A note on code balancing

- NSS uses a single code path for Intel® Microarchitectures Codenames Sandy Bridge, Ivy Bridge and Haswell
  - A good performance balance & code that runs on the three microarchitectures
  - Simple maintenance
- OpenSSL uses separate code paths
  - Achieving top performance on both architectures
  - The Haswell code path interleaves the encryption and the authentication
    - Gains 3% over the serial "CTR + GHASH" implementation that is used for Sandy Bridge and Ivy Bridge Microarchitectures
- The Haswell code path uses the (new) MOVBE instruction
  - Generates code that runs only on Haswell and onward
  - However: using MOVBE has no performance benefit over MOV+BSWAP or BSWAP+MOV
    - But artificially create "Haswell only" code

It is difficult to write code that is 100% optimal on Sandy Bridge, Ivy Bridge and Haswell Microarchitectures

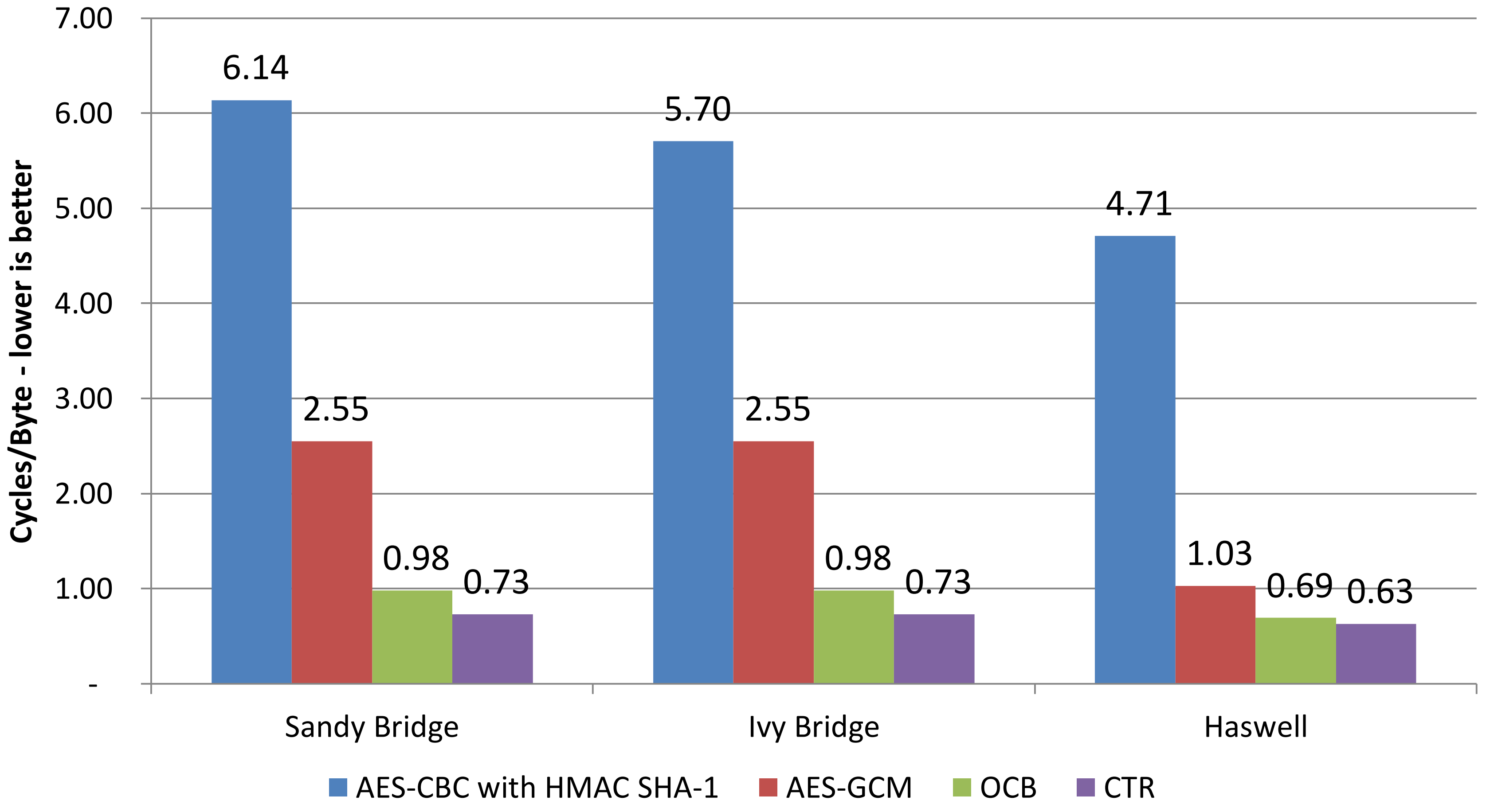
Encryption and Authentication breakdown

Processor Microarchitecture	Sandy Bridge	Ivy Bridge	Haswell
		Cycles/byte	
AES-GCM	2.53	2.53	1.03
GHASH	1.79	1.79	0.40
AES-CTR	**0.73**	**0.73**	**0.63**

- GHASH consumes ~ 70% of AES-GCM on Intel® Microarchitectures Codename Sandy Bridge and Ivy Bridge
- GHASH consumes ~40% on the latest Intel® Microarchitecture Codename Haswell
- Notes: the MAC computations are still significant
  - Limited by the (current) performance of PCLMULQDQ

Ultimate performance goal: AES-GCM at the same performance as CTR

Comparison to other AE schemes



Summary

- AES-GCM is the best performing Authenticated Encryption combination among the NIST standard options (esp. compared to using HMAC SHA-1)
  - Already enables in the open source libraries OpenSSL® and NSS
  - Performance keeps improving across CPU generations
  - The ultimate performance goal: AES-GCM at the performance of CTR+ ε
- With some luck, we might see significant deployment already in 2013, with leading browsers (Chrome/Firefox) offering AES-GCM as an option
- What about CAESAR competition candidates?
  - Performance on high end CPU's should be no less than the performance of AES-GCM (on the latest architecture)
  - Minimum - below 1 C/B. Target – **way below** 1 C/B
  - And now, you have a baseline to meet…

Proposing a minimum for the CAESAR competition candidates: at least as fast as AES-GCM on the high end CPU's

Thank you for listening

Questions? Feedback?

References

References

Software patches

- [PATCH] Efficient implementation of AES-GCM, using Intel's AES-NI, PCLMULQDQ instruction, and the Advanced Vector Extension (AVX),
- http://rt.openssl.org/Ticket/Display.html?id=2900&user=guest&pass=guest
- Efficient AES-GCM implementation that uses Intel's AES and PCLMULQDQ instructions (AES-NI) and the Advanced Vector Extension (AVX) architecture, https://bugzilla.mozilla.org/show_bug.cgi?id=805604
- [PATCH] Fast implementation of AES-CTR mode for AVX capable x86-64 processors, http://rt.openssl.org/Ticket/Display.html?id=3021&user=guest&pass=guest
- A patch for NSS: a constant-time implementation of the GHASH function of AES-GCM, for processors that do not have the AES-NI/PCLMULQDQ, https://bugzilla.mozilla.org/show_bug.cgi?id=868948

Papers:

AES-GCM

1. S. Gueron, Michael E. Kounavis: Intel® Carry-Less Multiplication Instruction and its Usage for Computing the GCM Mode (Rev. 2.01) http://software.intel.com/sites/default/files/article/165685/clmul-wp-rev-2.01-2012-09-21.pdf
2. S. Gueron, M. E. Kounavis: Efficient Implementation of the Galois Counter Mode Using a Carry-less Multiplier and a Fast Reduction Algorithm. Information Processing Letters 110: 549û553 (2010).
3. S. Gueron: Fast GHASH computations for speeding up AES-GCM (to be published).

AES-NI

5. S. Gueron. Intel Advanced Encryption Standard (AES) Instructions Set, Rev 3.01. Intel Software Network. http://software.intel.com/sites/default/files/article/165683/aes-wp-2012-09-22-v01.pdf
6. S. Gueron. Intel's New AES Instructions for Enhanced Performance and Security. Fast Software Encryption, 16th International Workshop (FSE 2009), Lecture Notes in Computer Science: 5665, p. 51-66 (2009).