Intel® QAT Engine for OpenSSL – Accelerating OpenSSL from Appliance to Public Cloud

Authors

Divya Pendyala

Joel D Schuetze

Heqing Zhu

1 Introduction

This document is intended for users who have incorporated Intel® QuickAssist Technology (Intel® QAT) hardware acceleration into their solutions and have recently or will be bringing those solutions to the cloud. We review how all users can benefit from either Intel® QAT hardware acceleration or cryptographic-specific instructions that were first available in 3rd Gen Intel® Xeon® Scalable processors.

Intel® Quick Assist Technology Engine for OpenSSL (Intel® QAT Engine for OpenSSL) or QAT_Engine is a software library that enables OpenSSL-based applications to accelerate cryptographic operations using either the Intel QAT hardware accelerator present on the target system or software acceleration available with 3rd Gen Intel Xeon Scalable processors or later.

Historically, Intel QAT hardware acceleration has been available in versatile form factors from a PCIe add-in card to a module within a system-on-chip (SoC) and more. With the introduction of the new crypto instructions, the QAT_Engine is now much more valuable as it takes advantage of either the QAT hardware accelerator or the new crypto instructions found on the target platform.

Starting in 2021, the 3rd Gen Intel Xeon Scalable processor was introduced with further advanced crypto instructions to accelerate network security workloads. With the Intel QAT Engine for OpenSSL, accelerating asynchronous mode applications like NGINX and HAProxy were optimized to provide significant performance improvements. As 3rd Gen Intel Xeon Scalable processors begin to deploy in the public cloud, the advancement of cryptographic acceleration brings a vast advantage for those placing workloads in public cloud such as Amazon Web Services, Google Cloud Platform, and Microsoft Azure.

Users having applications with cryptographic optimization using Intel QAT may need to change the software when moving it to public cloud because as of the date of this paper, Intel QAT is not present in VM platforms available at public cloud. This paper reviews the prerequisites and steps involved in installing the software crypto libraries, changing the software dependency from Intel QAT to the new crypto instruction set available on Intel Xeon Scalable processors, and enabling a qat_hw and qat_sw co-existing build to use a platform with both Intel QAT and 3rd Gen Intel Xeon Scalable processors.

This document is part of the Network Transformation Experience Kits.

Table of Contents

Figures

Tables

Document Revision History

Revision	Date	Description
001	December 2022	Initial release.
002	December 2022	Minor updates made to Section 5/ipp-crypto.

1.1 Terminology

Table 1. Terminology

Abbreviation	Description
AES-CBC	Advanced Encryption Standard Galois Counter Mode
AES-GCM	Advanced Encryption Standard Galois Counter Mode
DH	Diffie Hellman
DSA	Digital Signature Algorithm
ECDH	Elliptic-curve Diffie–Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
HKDF	Hashed message authentication code (HMAC)-based Key Derivation Function
Intel® AVX	Intel® Advanced Vector Extensions (Intel® AVX)
Intel® IPP Cryptography	Intel Integrated Performance Primitives Cryptography (Intel® IPP Cryptography)
Intel® QAT	Intel QuickAssist Technology (Intel® QAT)
ISA	Instruction Set Architecture
PKE	Public Key Encryption
PRF	Pseudorandom Function
QAT_Engine	Software implementing Intel® QuickAssist Technology Engine for OpenSSL (Intel® QAT Engine for OpenSSL)
RSA	Rivest-Shamir-Adelman (crypto algorithm)
SSL	Secure Sockets Layer
TLS	Transport Layer Security
USDM	User-Space DMA'able Memory
VAES	Vectorized Advanced Encryption Standard

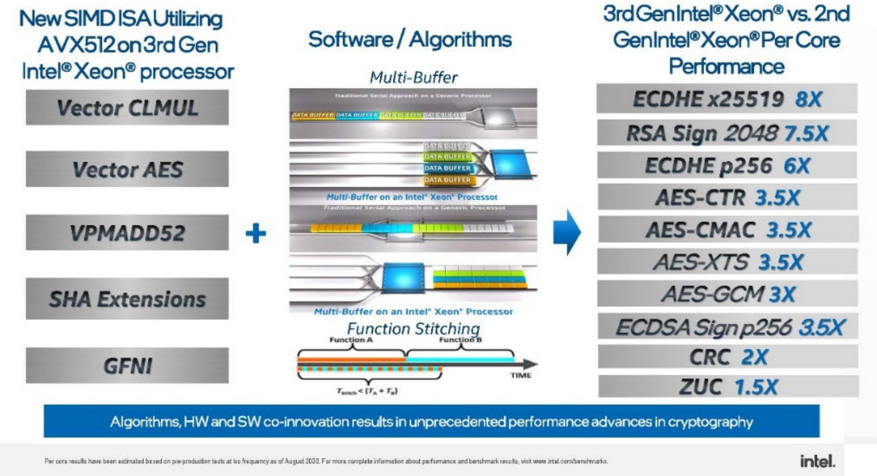
1.2 Reference Documentation

Table 2. Reference Documents

Reference	Source
Intel Xeon Scalable Platform Built for Most Sensitive Workloads	https://www.intc.com/news-events/press-releases/detail/1423/intel-xeon-scalable-platform-built-for-most-sensitive
Crypto Acceleration: Enabling a Path to the Future of Computing	https://newsroom.intel.com/articles/crypto-acceleration-enabling-path-future-computing
Intel® QAT Drivers download link	https://01.org/intel-quickassist-technology
GitHub for Intel® QAT	https://github.com/intel/QAT_Engine.git
GitHub for OpenSSL	https://github.com/openssl/openssl
GitHub for Intel® IPP Cryptography	https://github.com/intel/ipp-crypto
GitHub for Intel® Multi-Buffer Crypto for IPsec Library	https://github.com/intel/intel-ipsec-mb
Intel qat_hw and qat_sw Co-existence	https://github.com/intel/QAT_Engine/blob/master/docs/qat_common.md#qat-hw-and-qat-sw-co-existence

2 Crypto Acceleration on Intel 3rd Gen Intel Xeon Scalable Processors

Starting with the instruction set architecture (ISA), Intel introduced several enhancements designed to increase cryptographic performance significantly. For example, new ISA support for "big number" multiplication often found in public-key ciphers, Intel® Advanced Vector Extensions 512 (Intel® AVX-512) Integer Fused Multiply Add (AVX512_IFMA). Combined with software optimization techniques such as multi-buffer processing, these instructions provide significant performance improvements for RSA and elliptic curve cryptography. Multi-buffer is an innovative and efficient technique for processing multiple independent data buffers in parallel for cryptographic algorithms. Vectorized AES (VAES) and vectorized carryless multiply instructions increase performance for AES symmetric encryption. With the newly added Intel® Secure Hash Algorithm Extensions (Intel® SHA Extensions), SHA-256 gets a boost in performance.



New SIMD ISA Utilizing AVX512 on 3rd Gen Intel® Xeon® processor

- Vector CLMUL
- Vector AES
- VPMADD52
- SHA Extensions
- GFNI

Software / Algorithms

Multi-Buffer

Function Stitching

3rd Gen Intel® Xeon® vs. 2nd Gen Intel® Xeon® Per Core Performance

- ECDHE x25519 8X
- RSA Sign 2048 7.5X
- ECDHE p256 6X
- AES-CTR 3.5X
- AES-CMAC 3.5X
- AES-XTS 3.5X
- AES-GCM 3X
- ECDSA Sign p256 3.5X
- CRC 2X
- ZUC 1.5X

Algorithms, HW and SW co-innovation results in unprecedented performance advances in cryptography

Per core results have been estimated based on pre-production tests at iso frequency as of August 2020. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

+
Traditional Serial Approach on a Generic Processor
DATA BUFFER
DATA BUFFER
DATA BUFFER
DATA BUFFER
Multi-Buffer on an Intel® Xeon® Processor
Traditional Serial Approach on a Generic Processor
Multi-Buffer on an Intel® Xeon® Processor
Function A
Function B
TIME
$T_{stitch} < (T_A + T_B)$
intel.

Figure 1. Performance Boost from Cryptographic Algorithms on 3rd Gen Intel Xeon Scalable Processor

These additional instructions VPMADD52 - vector instruction that does integer multiply accumulate, vAES - vector version of the Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI) instructions, vCLMUL - vector version of the CLMUL instruction, and Intel® Secure Hash Algorithm – New Instructions (Intel® SHA-NI). The combination of vAES and vCLMUL on the wide registers that are available on the Intel AVX-512 further speed up AES modes such as AES-CTR and AES-CBC. VPMADD2 is targeted at significantly reducing the instructions needed to generate public/private keys as part of an RSA-2K sign operation. Intel SHA-NI looks to improve hashing functions used in cryptographic protocols such as SSL/TLS as well helping with data deduplication in storage workloads.

3 Intel QuickAssist Technology Engine for OpenSSL

Intel QAT Engine for OpenSSL (QAT_Engine) is a software package that supports acceleration for both hardware and optimized software based on vectorized instructions. The advancement in cryptographic acceleration in 3rd Gen Intel Xeon Scalable processors provides users more options to accelerate their workloads. The QAT_Engine now supports the ability to accelerate the standard OpenSSL using basic Intel instruction set to either the hardware acceleration path (via the Intel QAT hardware (qat_hw) path) or via the optimized software path (qat_sw lib). This document details the steps involved in changing the software dependency from the Intel QAT (qat_hw) path to using the new crypto instruction set (qat_sw) path.

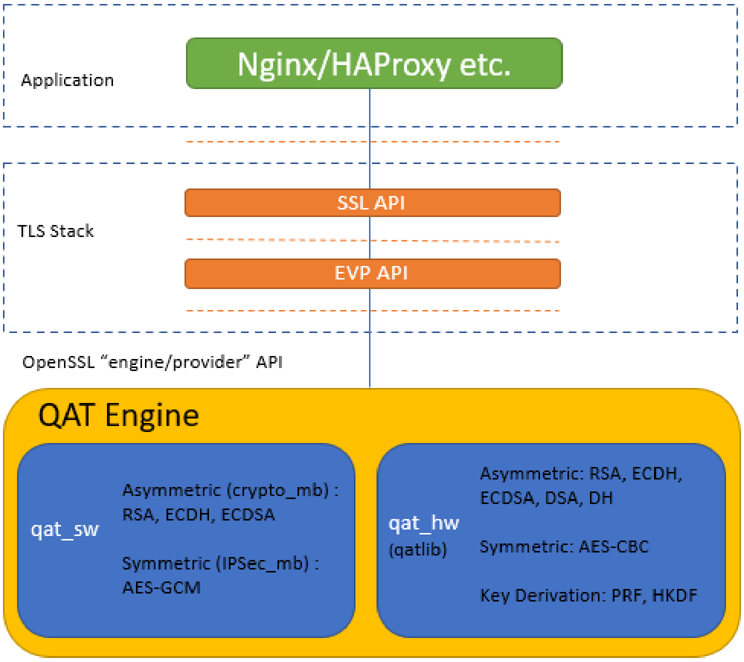

Application
Nginx/HAProxy etc.
TLS Stack
SSL API
EVP API
OpenSSL "engine/provider" API
QAT Engine
qat_sw
Asymmetric (crypto_mb) : RSA, ECDH, ECDSA
Symmetric (IPSec_mb) : AES-GCM
qat_hw
(qatlib)
Asymmetric: RSA, ECDH, ECDSA, DSA, DH
Symmetric: AES-CBC
Key Derivation: PRF, HKDF

Figure 2. Intel QuickAssist Technology Engine for OpenSSL Stack on 3rd Gen Intel Xeon Scalable Processor

Figure 2 illustrates the high-level software architecture of the QAT_Engine. Applications such as NGINX and HAProxy are common applications that interface to OpenSSL. OpenSSL is a toolkit for TLS/SSL protocols that, starting with version 1.1.0, includes a modular system to plugin device-specific engines. As mentioned earlier, within the QAT_Engine are two separate internal entities by which acceleration can be performed. Depending on your particular use case, the QAT_Engine can be configured to meet your specific acceleration needs. Both the hardware and software requirements are explained followed by detailed instructions on how to install QAT_Engine.

4 Intel QuickAssist Technology (Intel QAT)

Starting with Intel 3rd Gen Intel Xeon Scalable processors, Intel QAT can be a PCIe add-in card or integrated on System-on-Chip (SoC). Intel QAT is available as a PCIe add-in card, in Intel Xeon D processors integrated in SoC, and previously in the Intel chipset for Intel Xeon processors. To use Intel QAT acceleration with the QAT_Engine, qat_hw is the path to use to optimize software with cryptography.

4.1 Prerequisites for qat_hw

The following are the prerequisites for setting up the qat_hw path.

- qat_hw path requires Intel Xeon processor with Intel® C62X Series Chipset
- Intel Atom® processor, or Intel Xeon Scalable processor, or Intel Xeon D processor
- Intel® Communications Chipset 8925 to 8955 Series
- Intel QuickAssist Technology driver for the operating system

  Note: Intel QAT drivers can be downloaded from the following link: https://01.org/intel-quickassist-technology
- Intel QAT driver installation using the instructions from the Getting Started Guide

4.2 Configuration Steps for Intel QAT Engine for OpenSSL with qat_hw

- Build OpenSSL

```
git clone https://github.com/openssl/openssl.git
cd /openssl
./config [options] -Wl,-rpath,\${LIBRPATH}
make
make install
```

- Build QAT_Engine

```
git clone https://github.com/intel/QAT_Engine.git
cd /QAT_Engine
./configure --with-qat_hw_dir=/QAT
make
make install
```

The Intel QAT Engine for OpenSSL was cloned to its own location at the root of the drive: `/`.

The Intel QAT Driver was unpacked within `/QAT` and using the USDM component.

Prebuilt OpenSSL (both library and developer RPM packages) are installed in the system and the OpenSSL version is in the 1.1.1 series.

5 Software-Based Acceleration (qat_sw)

When a platform does not have Intel QAT configured, but has a 3rd Gen Intel Xeon Scalable processor, you can choose the qat_sw path to use the new crypto instructions in the Intel AVX-512 to accelerate cryptography. This software-based acceleration qat_sw path is incorporated into the Intel QAT Engine for OpenSSL, a dynamically loadable module that uses the OpenSSL ENGINE framework, allowing administrators to add this capability to OpenSSL without having to rebuild or replace their existing OpenSSL libraries.

Following are the steps to change from qat_hw path to qat_sw path.

1. Prerequisites for qat_sw
   - qat_sw requires a software tool chain that supports OpenSSL 1.1.1 or OpenSSL 3.0
   - Intel® Integrated Performance Primitives Cryptography (Intel® IPP Cryptography) (for Asymmetric PKE) - ipp-crypto 2021.5
   - Intel® Multi-Buffer crypto for IPsec Library (for AES-GCM acceleration) - intel-ipsec-mb v1.2
   - Operating system: Ubuntu 20.04.2 LTS
2. Configuration steps to change from qat_hw path to qat_sw path
   - Build OpenSSL (skip if installed already)

     ```
     git clone https://github.com/openssl/openssl.git
     cd /openssl
     ./config [options] -Wl,-rpath,\${LIBRPATH}
     make
     make install
     ```

   - Build ipp-crypto for Asymmetric PKE

     ```
     git clone https://github.com/intel/ipp-crypto
     cd ipp-crypto
     Ensure you are building against a fixed release of the code, and not the development branch.
     At the time of this writing, the latest release was ippcp_2021.6
     git checkout ippcp_2021.6
     cd sources/ippcp/crypto_mb
     cmake . -Bbuild -DCMAKE_INSTALL_PREFIX=/usr
     cd build
     make -j
     sudo make install
     ```

   - Build ipsec_mb for AES-GCM

     ```
     git clone https://github.com/intel/intel-ipsec-mb.git
     cd intel-ipsec-mb
     Ensure you are building against a fixed release of the code, and not a development branch.
     At the time of this writing, the latest release was v1.3
     git checkout v1.3
     make -j
     ```

   - `make install NOLDCONFIG=y`Build QAT_Engine

     ```
     git clone https://github.com/intel/QAT_Engine.git
     cd /QAT_Engine
     ./configure --enable-qat_sw --disable-qat_hw
     make
     make install
     ```

     `--enable-qat_sw` checks the `crypto_mb` and `IPSec_MB` libraries in their respective default paths or in the path provided in the config flag `--with-qat_sw_install_dir`. If any of the libraries are not installed, then their corresponding algorithm support is disabled (`cryto_mb` library for PKE algorithms and `IPSec_mb` library for AES-GCM).

6 Co-existence of qat_hw and qat_sw

Intel QAT Engine for OpenSSL supports a platform with Intel Xeon processors with cryptography ISA and Intel QAT. You can use qat_hw and qat_sw co-existence build with both qat_hw and qat_sw dependent libraries (QAT driver, crypto_mb, and ipsec_mb) linked in the `qatengine.so` library.

This support can be enabled at engine build time when both qat_hw flag `--with-qat_hw_dir=/path/to/QAT_Driver` and qat_sw flag `--enable-qat_sw` configured together in the build configure option.

- Prerequisites for qat_sw and qat_hw

Install and build both qat_hw driver and qat_sw libraries

- Build QAT_Engine

```
git clone https://github.com/intel/QAT_Engine.git
cd /QAT_Engine
./configure --with-qat_hw_dir=/QAT --enable-qat_sw
```

If the platform has support for both qat_hw and qat_sw, the default behavior is to accelerate asymmetric algorithms and symmetric chained ciphers using Intel QAT (qat_hw) and Symmetric GCM Ciphers using cryptography ISA (qat_sw) in Intel Xeon processors. If the platform does not have Intel QAT hardware support, then it uses qat_sw acceleration for qat_sw asymmetric algorithms that are supported in the QAT_Engine.

The default behavior can be changed using the corresponding algorithm's enable flags (for example, `--enable-qat_sw_rsa`, `--enable-qat_hw_gcm`) in which case the individual algorithms enabled (either qat_hw or qat_sw) in the build configure are accelerated.

Intel QAT Engine for OpenSSL supports a runtime mechanism that dynamically chooses the qat_hw path or qat_sw path or both paths for each algorithm, using qat_hw and qat_sw dependent libraries linked in a single QAT Engine. It can be accomplished through two ENGINE ctrl commands: `HW_ALGO_BITMAP` and `SW_ALGO_BITMAP`. Further details on the bitmap of each algorithm and how to enable them can be found in the following link:
https://github.com/intel/QAT_Engine/blob/master/docs/qat_common.md#qat-hw-and-qat-sw-co-existence

7 Summary

Intel QuickAssist Technology (Intel QAT) is a mature technology that has been in the market for more than a decade. It has been adopted by cloud, networking, storage, and big data customers for three main services: Cryptographic Cipher and Hash, Public Key Crypto, and Compression/Decompression. Intel QuickAssist Technology Engine for OpenSSL (QAT_Engine) supports acceleration using both Intel QAT hardware as well as Intel Xeon processors with vectorized instructions for offloading cryptographic operations interfacing with OpenSSL.

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.