WHITE PAPER
Securing the Enterprise with Intel® AES-NI
Intel® Advanced Encryption Standard New Instructions (AES-NI)

Securing the Enterprise with Intel® AES-NI

Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI)

Leslie Xu
Intel Corporation

Acknowledgements to Jeffrey Casazza, Michael Kounavis, Shihjong Kuo, and Woody Cohn for their contribution.

September 2010
Version 2.0

EXECUTIVE SUMMARY

Our world today has insatiable demand for technology that can find, process, and communicate information and data in business environments, as well as in our personal lives. Protection of intellectual property, personal identities, and other sensitive information is more important than ever for data on the move and at rest.

A large part of this protection is achieved through cryptography. Cryptography is the science of secret codes—transforming data from ordinary readable forms into unintelligible forms, thereby enabling the confidentiality of communication through an insecure or public/shared channel. Cryptography protects data against unauthorized parties by preventing its unauthorized use or alteration. The challenge is that traditionally, cryptography has been complex and computationally costly to execute.

Generally speaking, a cryptographic system employs a mathematical or algorithmic process to transform readable plain text to coded "cipher text" and then convert that cipher text back to plain text. The algorithms used in the encryption/decryption processes are referred to as ciphers. The operation of a cipher is often controlled by a key or a set of keys. A number of factors, such as confidentiality, integrity, authenticity, and performance determine the end-user benefits of different encryption standards.

Why is cryptography hot in the marketplace today, especially in the enterprise? For starters, over 345 million records containing sensitive personal information have been involved in security breaches in the United States since January 2005.[1] The rate is accelerating and the attacks are more complex and harder to detect. There is a shift from random attacks on multiple computers to targeted attacks on a few high-value bank or government systems with sensitive financial and personally identifiable information. In the highly virtualized environment of computing today, several virtual machines share the same hardware resources. The hardware resources need more secure protection as there are more eggs in one basket. Encryption provides a deep enough defense so that even if the systems are compromised and information is lost, the information remains unusable through symmetric and asymmetric crypto schemes. Encryption also provides data protection increasingly important for compliance with Health Insurance Portability and Accountability Act (HIPAA); Sarbanes-Oxley Act (SOX), which affects U.S. companies; and Payment Card Industry (PCI) regulations. Note that HIPAA requires encryption only when data is transmitted over the public Internet; but the HiTECH act adds extensive breach notification requirements and enforcement capabilities to HIPAA when data is not encrypted.

Table of Contents

2. Introduction 2

2.1. Definitions 3

3. The Advanced Encryption Standard (AES) 5

4. Introducing Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI) 5

5. AES Usage Models 6

5.1. Secure Transactions 6

5.1.1. HTTPS in the Cloud 7

5.1.2. Internet Protocol Security (IPsec) 7

5.2. Enterprise Applications 8

5.3. Full Disk Encryption (FDE) 8

6. Performance Implications 9

6.1. Secure Transactions Performance 9

6.2. Application-level Encryption Performance 10

6.3. Full Disk Encryption 10

7. Implementing in Applications . . 11

7.1. Operating Systems 11

7.2. Libraries 11

7.2.1. Intel® Integrated Performance Primitives Library 11

7.2.2. Java* Cryptography Extensions (JCE) 11

7.2.3. RSA* BSAFE* 12

7.2.4. Crypto++ 12

7.2.5. OpenSSL* 12

7.2.6. Linux* Kernel 12

7.3. Compilers 12

8. Conclusion 12

One popular encryption standard is the Advanced Encryption Standard (AES). Adopted by the U.S. government in 2001, it is widely used today across the software ecosystem to protect network traffic, personal data, and corporate IT infrastructure. AES applications include secure commerce, data security in database and storage, secure virtual machine migration, and full disk encryption. According to an IDC Encryption Usage Survey, the most widely used applications are corporate databases and archival backup.[2] Full-disk encryption is also receiving lots of attention.

Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI) is a set of seven new instructions in the Intel® Xeon® processor 5600 series (formerly codenamed Westmere-EP). Four instructions accelerate encryption and decryption. Two instructions improve key generation and matrix manipulation. The seventh aids in carry-less multiplication. By implementing some complex and costly sub-steps of the AES algorithm in hardware, Intel AES-NI accelerates execution of the AES-based encryption. The result is faster, more secure encryption, which makes the use of encryption feasible where not before.

This paper will address AES and Intel AES-NI in detail, followed by an examination of three usage models, performance improvement implications, and the cryptographic libraries that independent software vendors (ISVs) can use to replace basic AES routines with the Intel AES-NI optimizations.

2. Introduction

According to the IDC Encryption Usage Survey, since 2005 more than 90 million consumers have been notified of potential security breaches regarding personal information.[3] Turning on the laptop in the morning brings up a Wi-Fi network property window that indicates AES-CCMP data encryption with enterprise authentication. Working on an intranet or making a purchase on a secure Web site, brings up a lock icon on your browser. This indicates a secure connection provided by Secure Socket Layer (SSL). SSL is a cryptographic protocol that provides security and data integrity for communication over networks such as the Internet. Secure Socket Layer (SSL) and the more recent Transport Layer Security (TLS) protocol encrypt the segments of network connections end-to-end.

Clearly, security surrounds us daily at a personal (client) level, but what about in the enterprise and its servers? Many government and corporate servers contain large quantities of personally identifiable information, as well as financial information, that is distributed to clients upon request. This makes encryption at the server level critically important, particularly since the growth rate of malicious code (malware) used to attack computers of all kinds continues to accelerate.

Equally concerning, computer attacks are getting more complex and harder to detect. What's more, there is an ongoing shift in the type of person initiating the attacks. In the past, many perpetrators were after notoriety, to show off their technical skill and infect the most number of computers. This type of attacker is being replaced by attackers motivated by money and involved with organized crime. Their goal is not always to infect millions of computers, but instead to stealthily infect a few high-value targets. These targets could be banks and institutions which have access to financial and personal information. Encryption provides a good defense of last resort. Even if systems are compromised and the information is taken, encryption can keep it unusable.

Whether it's classified government data stored on a laptop hard drive or data center security breaches involving customers' social security numbers, the sheer increase in the number of incidents, as well as their impact, has led to legislative initiatives in a growing number of states that mandate the use of encryption technology for sensitive information.[4] Many government agencies are requiring disclosure of security breaches, and federal legislation requiring disclosure has been proposed. Industries are also increasing scrutiny of their security procedures. For many, encryption provides increasingly important data protection for helping remain compliant with HIPAA, SOX, PCI and other regulations. Protecting sensitive data increases customer trust and loyalty, reduces legal liability, and helps meet regulatory requirements for data security.

Below are some examples of regulations that require or strongly encourage encryption and the penalties for noncompliance.

HIPAA 4 Health Insurance Portability and Accountability Act of 1996:

- Information systems housing patient health information must be protected from intrusion.
- When information flows over open networks, some form of encryption must be utilized.
- Potential penalty for noncompliance: 10-year prison sentence and $100 fine per incident with a maximum of $25,000 per year.

HiTECH Act February 17, 2010:

- Extended the reach of the HIPAA Privacy and Security Provisions to business associates (BAs).
- Covered entities must provide major media notice whenever a breach involves more than 500 individuals in a state.
- If personal health information is rendered "unusable, unreadable or indecipherable," no notification is required.
- A major security breach that results in actual damages can lead to class action lawsuits, regulatory action, drop in stock price and damage to reputation and customer relationships.
- Sets encryptions standards for "unusable, unreadable or indecipherable" data at rest and in motion.

Sarbanes-Oxley Act7 (SOX):

- Requires all publicly held U.S. companies to comply with strict information technology guidelines to protect the integrity and confidentiality of financial information.
- ISO/IEC 27002 (an information security standard) defines best practices for SOX-related security controls and explicitly suggests the use of encryption.
- Potential penalty for noncompliance: 10-year prison sentence and $15,000,000 fine.

Payment Card Industry (PCI) Data Security Standard:

- Requires Primary Account Numbers (PAN) and/or credit card numbers to be encrypted while at rest.
- Members of the organization include Visa, MasterCard, American Express, Discover, and others.
- Members have the right to quit accepting credit card transactions from noncompliant institutions and assess fines of $500,000.

Increasingly, protecting sensitive data is becoming mandatory, as is the need to use encryption and encryption technology.

2.1. Definitions

The following are a few definitions needed to better understand this paper.

AES: Advanced Encryption Standard. A slightly modified version of the Rijindael algorithm, AES is an encryption standard adopted by the U.S. government in 2001. AES is displacing the older, less secure 3DES (Data Encryption Standard) with 112 or 168 bit key lengths. AES is a block cipher, which means it works on fixed-length groups of bits, which are called blocks.

Algorithm: A set of ordered steps (a mathematical formula) to accomplish a task. In encryption, that task would be to translate normal data into secret characters (encryption) or translate secret characters back into readable data (decryption).

Asymmetric cryptographic algorithm: A scheme that uses a different key for encryption than for decryption. A user knowing one key of an asymmetric key pair can encrypt data, but cannot decrypt data encrypted with that key. This is a another encryption method than systems that use the same key to encrypt and decrypt data, mostly used to encrypt small amounts of data, like symmetric encryption keys.

Cipher: An algorithm to encrypt or decrypt information exchanged by two or more parties to maintain a private information exchange.

Ciphertext: Data translated by an algorithm into an encrypted form. Cipher text is unreadable until it has been decrypted back into its original data form.

Decryption key: The key that decrypts the message. Symmetric keys use the same key to encrypt and decrypt, asymmetric keys do not.

Encryption key: The key that encrypts the message.

Hash function: A well-defined procedure or mathematical function that converts a block of data into a fixed-size bit string

(the hash value) so that a change to the data will change its hash value. The data being encoded is referred to as the "message." The hash value is often called the "digest." Most cryptographic hash functions take a string of any length and produce a fixed-length hash value.

IPsec (Internet Protocol security): A framework of open standards for protecting communications over Internet Protocol (IP) networks through the use of cryptographic security services. (Implemented in layer three of the Open Systems Interconnection (OSI) protocol standard in the form of crypto software for end-to-end Ethernet connection.)

MAC: Message Authentication Code. A short piece of information used to authenticate a message. A MAC algorithm is often referred to as a keyed (cryptographic) hash function. MACs are different from digital signatures in that MAC values are generated and verified using the same secret key. To do that, the message sender and receiver must agree on the same key before initiating communications. This is similar to symmetric encryption.

NIC: Network Interface Controller, also known as Ethernet adapter. NICs may contain hardware to process traffic and offload IPsec traffic to Intel AES-NI for the processor.

Plaintext: The original message or original file. This is what you get after a file or message has been encrypted and then decrypted.

RSA: Rivest-Shamir-Adleman encryption algorithm. An asymmetric cryptographic algorithm (public and private key; compared to AES where encryption and decryption is done with a single key). The pre-master secret exchanged during RSA leads to a pre-master key, from which both HMAC (keyed-hash authentication code) key and AES keys are dynamically derived.

SHA: Secure Hash Algorithm. A set of cryptographic hash functions. SHA was designed by the National Security Agency (NSA) and published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard. (NIST is a U.S. federal government agency which establishes standards and guidelines for private and public sector purposes.) SHA-1 produces a 160-bit digest from a message with a maximum length of (264 – 1) bits.

SSL: Secure Socket Layer. A cryptographic protocol which provides secure communications over networks. Implemented in layer six (the presentation layer) of OSI for use in Web application-level secure transactions.

Software side-channel attacks: An attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms. These attacks typically use timing, thermal, noise, cache contents, electromagnetic radiation, power consumption, or other indirectly gathered information about a cryptographic operation to reduce the key space that must be searched to identify the key and break the system.

Symmetric cryptographic algorithm: A scheme that uses closely related, often identical, cryptographic keys for both encryption and decryption. Asymmetric key algorithms are generally hundreds to thousands of times slower than symmetric key algorithms.

TPM: Trusted Platform Module. A specialized chip that can be installed on the system board of a computer to add cryptographic functions and support cryptographic calculations. It works with supporting firmware and software to prevent unauthorized access to the system. The TPM contains a hardware engine which only performs up to 2048-bit RSA encryption/decryption. The TPM uses its built-in RSA engine during digital signing and key wrapping operations.

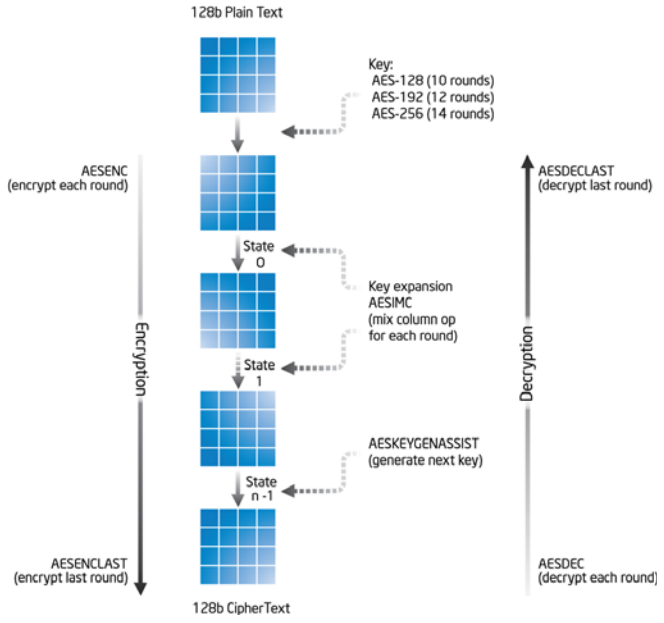

Figure 1. Intel® AES-NI accelerates sub-steps of the AES algorithm.

3. The Advanced Encryption Standard (AES)

AES, a slightly modified version of the Rijindael algorithm, is an encryption standard adopted by the U.S. government in 2001. AES is displacing the older, less secure 3DES (Data Encryption Standard) with 112 or 168 bit key length. AES is a block cipher, which means that it works on fixed-length group of bits, which are called blocks.

Figure 1 diagrams the flow of the AES algorithm. AES uses a fixed 4x4 block size of 128 bits and a variable key length. Depending on the key length (128, 192, or 256 bits), either 10, 12, or 14 rounds of transformation are required to produce the final cipher text. The original plaintext (4x4 bytes) gets encrypted with a key specific for the first cipher round which produces an intermediate result or state. The resulting intermediate ciphertext is fed into the next round with a second round key; and so on until the defined number of rounds is complete (see Figure 1). The final round is the same as previous rounds except it excludes a MixColumn operation where the four bytes of each column are combined using an invertible linear transformation (this is accomplished through multiplication with a fixed polynomial).

AES has several different modes of operation. For block cipher modes of operation, NIST recommends electronic codebook (ECB), cipher block chaining (CBC), counter (CTR), cipher feedback (CFB), and output feedback (OFB). The most basic AES mode ECB is simple and parallel, but is not considered secure because it is vulnerable to statistical attacks (a type of attack that employs statistical methods applied to plaintext/ciphertext pairs to distinguish some part of the cipher from a random permutation). For this reason, CBC is the most commonly used mode of operation. It is serial in nature. CTR is a common parallel mode, but does not have authentication. Galois counter mode (GCM) combines CTR with an authentication tag, which is an alternative to the Secure Hash Algorithm (SHA).

For more information on the AES encryption standard see FIPS PUBS 197.[5]

4. Introducing Intel® AES-NI

Intel AES-NI (Advanced Encryption Standard New Instructions) is a set of new instructions in the Intel® Xeon® processor 5600 series (formerly codenamed Westmere-EP). Intel AES-NI implements in the hardware some sub-steps of the AES algorithm. This speeds up execution of the AES encryption/decryption algorithms and removes one of the main objections to using encryption to protect data: the performance penalty.

To be clear, the Intel Xeon processor 5600 series doesn't implement the entire AES application. Instead, it accelerates just parts of it. This is important for legal classification purposes because encryption is a controlled technology in many countries. Intel AES-NI adds six new AES instructions, four for encryption and decryption, one for the mix column, and one for generating next round text. These instructions speed up the AES operations in the rounds of transformation and assist in the generation of the round keys.

Intel AES-NI also includes a seventh new instruction: CLMUL. This instruction could speed up the AES-GCM and binary Elliptical Curve Cryptography (ECC), and assists in error correcting codes, general purpose cyclic redundancy checks (CRCs), and data de-duplication. It particularly helps in carry-less multiplication, also known as "binary polynomial multiplication." This is the mathematical operation of computing the product of two operands without generating or propagating carries. Such multiplications are an essential step in computing multiplications in binary Galois fields. Intel AES-NI includes Intel's carry-less multiplication instruction. Algorithms can use CLMUL to compute the Galois Hash, the underlying computation of the GCM. CLMUL speeds up execution of GCM by computing the carry-less multiplication of two 64-bit operands.

Looking back at Figure 1, it shows four of the new instructions at work encrypting and decrypting rounds from a 128 bit plain text to 128 bit cipher text and vice versa. During each round, there are two instructions that assist in generating the follow-on key. The AESENC instruction encrypts each round and AESENCLAST encrypts the last round. In reverse (decryption), AESDEC decrypts each round and AESDECLAST decrypts the last round. Another instruction, AESIMC, does the mix column operation for each round and AESKEYGENASSIST generates the next key. The keys can be 128, 192, or 256 bit. Remember, all these computations are being done by the hardware, providing a significant speedup: 4x in CBC encrypt in serial mode and more than 14x in parallel modes of operation (see Section 6.1 for details).

Besides the performance benefit of Intel AES-NI, its execution of instructions in hardware provides some additional security in helping prevent software side-channel attacks. Software side channels are vulnerabilities in the software implementation of cryptographic algorithms. They emerge in multiple processing environments (multiple cores, threads, or operating systems). Cache-based software side-channel attacks exploit the fact that software-based AES has encryption blocks, keys, and lookup tables held in memory. In a cache collision-timing side-channel attack, a piece of malicious code running on the platform could seed

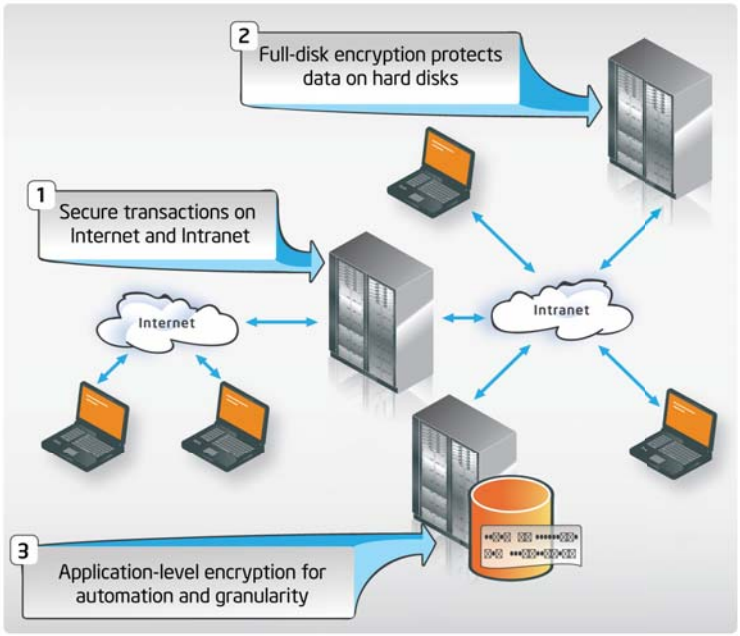

Figure 2. Three AES usage models.

the cache, run cryptographic operations, then time specially crafted memory accesses to identify changes in the cache. From these changes, the attack could determine portions of the cryptographic key value. For example, by measuring the time it takes for a given cryptographic operation, an attacker may be able to determine that the uppermost bit of a key is a "0." Knowing that single bit cuts in half the key space that must be searched to identify the complete key value. More effective side-channel attacks reduce the key space significantly (i.e., they may identify half the bits in the key).

Since Intel AES-NI is hardware-based, it has no need for lookup tables and the encryption blocks are executed in hardware within the microprocessor. This enables implementations of AES that use Intel AES-NI to address software side-channel attacks.[6] In addition, these instructions make AES simple to implement, with reduced code size. This helps reduce the risk of inadvertent introduction of security flaws, such as difficult-to-detect side channel leaks. Furthermore, the acceleration provided by Intel AES-NI can allow the system to execute larger key sizes, thus making data transfers more secure.

Intel AES-NI performance and strength benefits will be available to customers who buy servers based on the Intel Xeon processor 5600 series and software that has been optimized for the new instructions.

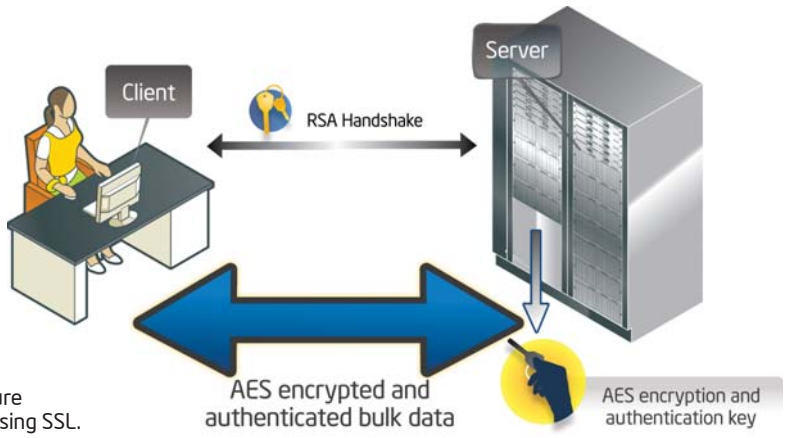

Figure 3. Secure transactions using SSL.

For more information on the AES new instructions, see: http://software.intel.com/file/24917.

For more information on the CLMUL instruction and its handling of carry-less multiplication, see: http://software.intel.com/en-us/articles/carry-less-multiplication-and-its-usage-for-computing-the-gcm-mode.

5. AES Usage Models

This section identifies three main usage models (see Figure 2) for Intel AES-NI: network encryption, full-disk encryption (FDE), and application-level encryption.

Networking applications use encryption to protect data in flight with protocols encompassing SSL, TLS, IPsec, HTTPS, FTP, and SSH. This section focuses on HTTPS and IPsec, as well as FDE and application-level models that use encryption to protect data at rest.

In all three of these models, improved performance is gained from using Intel AES-NI. Such performance improvements can enable the use of encryption where it might have otherwise been impractical due to performance impact.

5.1. Secure Transactions

In today's highly networked world, Web servers, application servers, and database backend all connect via an IP network

through gateways and appliances. SSL is typically used to deliver secure transactions over the network. It's well known for providing secure processing for banking transactions and other ecommerce, as well as for enterprise communications (such as an intranet).

In the secure transaction depicted in Figure 3, the user first reaches a Web page with a URL starting with https://. HTTPS combines Hypertext Transfer Protocol (HTTP) and a cryptographic protocol to create a secure channel over an insecure network. HTTP operates at the top layer (application layer of the OSI protocol standard), but HTTPS leverages the SSL protocol in the presentation layer. Most new servers deployed that support SSL also support AES as a cipher choice. However, for a browser transaction with HTTPS, the cipher suite choice is heavily influenced by the client's operating system (OS), not the browser or server support. AES is the predominant choice in Microsoft* Windows* 7 and Vista* and also Linux* operating systems. The large Microsoft Windows XP installed base currently uses the older, less secure RC4-MD5 cipher suite for SSL transactions. A server picks the strongest cipher suite and hash function that both the server and the client support, and it notifies the client of the decision.

Once the client contacts the server to start a transaction (back to Figure 3), an SSL transaction starts with a handshake between the client and server using the RSA encryption algorithm. RSA requires the server to send a public key to the client, followed by the client sending the server a pre-master secret which the server decrypts. Because RSA is an asymmetric algorithm, the client will have to use a different key to decrypt the data. AES uses a similar handshake authentication to the RSA handshake, but is a symmetric algorithm, so the same key will encrypt and decrypt. After the AES handshake, the bulk exchange of authenticated data via AES starts.

Where Intel AES-NI provides a real opportunity is in reducing the computation impact (load) for those SSL transactions that use the AES algorithm. There is significant overhead in establishing secure communications, and this can be multiplied by hundreds or thousands depending on how many systems want to concurrently establish secure communications with a server. Think of your favorite online shopping site during the holiday season. Integrating Intel AES-NI would improve performance by reducing the computation impact of all these secure transactions.

5.1.1. HTTPS in the Cloud

While HTTPS connections are typically used for payment transactions on the World Wide Web and sensitive transactions in corporate information systems, they're also popular for email, portals, and collaborative software over intranets and the Internet. With the growing popularity of cloud services like Google* Apps and Windows Live*, secure HTTPS connections are getting increased attention—and use.

The growth in cloud services is putting enormous amounts of user data on the Web. To protect users, operators of public or private clouds must ensure the privacy and confidentiality of each individual's data as it moves between client and cloud. This means instituting a security infrastructure across their multitude of service offerings and points of access. For these reasons, the amount of data encrypted, transmitted, and decrypted in conjunction with HTTPS connections is predicted to grow as clouds proliferate.[7]

For cloud providers, the performance and responsiveness of transactions, streaming content, and collaborative sessions over the cloud are all critical to customer satisfaction. Yet the more subscribers cloud services attract, the heavier the load placed on servers. This makes every ounce of performance that can be gained anywhere incredibly important. Intel AES-NI and its ability to accelerate the performance of encryption/decryption can play a significant role in helping the cloud computing movement improve the user experience and speed up secure data exchanges.

5.1.2. Internet Protocol Security (IPsec)

While SSL is normally used between Web applications and services at layer seven of OSI, IPsec connections use known peers at layer three as shown in Figure 4.

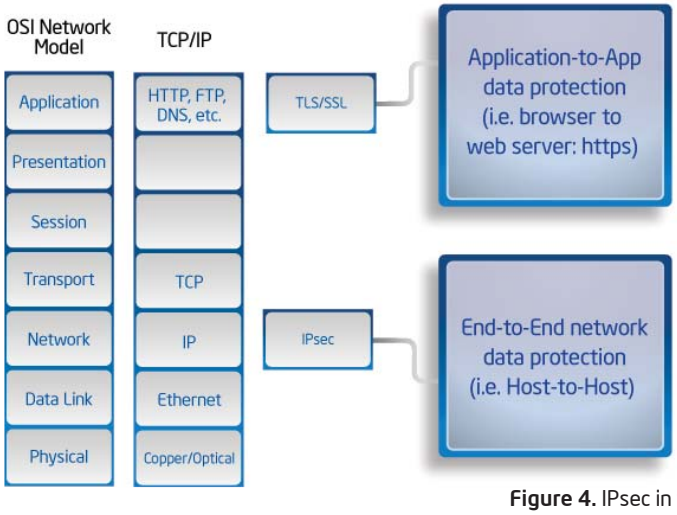

Figure 4. IPsec in the OSI stack.

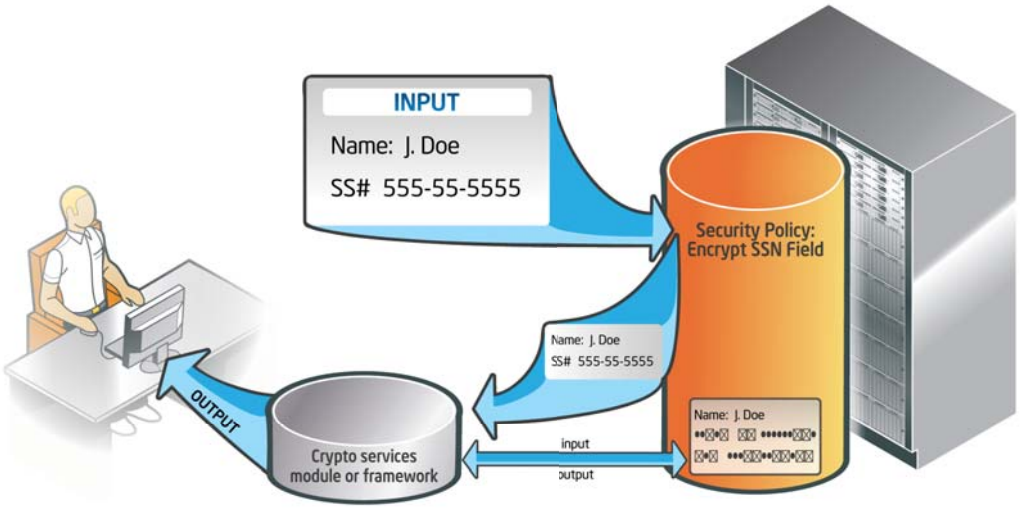

Figure 5. Application-level encryption AES usage model.

A specific policy for a chosen set of ports and encryption/decryption is done with socket library (a low-level programmer's interface that allows clients to set up a TCP/IP connection and communicate directly to servers) or OS library.

IPsec can be also offloaded for acceleration to an Ethernet NIC, such as the Intel® 82599 10 Gigabit Ethernet Controller or Intel® 82576 Gigabit Ethernet Controller. When only a few connections are needed, the NIC offload is likely the preferred solution for IPsec. When there are many connections, Intel AES-NI is a better solution (given the limitations in existing offload engines around number of connections) or a hybrid approach, which can either offload to NIC or divert to OS crypto library for Intel AES-NI acceleration.

A typical scenario for using IPsec is protecting traffic between a corporate office and a remote office. Traffic in the remote office could be 12 to 15 connections, but connections in the corporate office could be 12,000 connections. For the remote office, a pure offload approach makes sense. For the corporate office, a hybrid approach improves performance. Whenever AES encryption/decryption comes into play, deploying servers based on Intel Xeon processor 5600 series and software that has been optimized for Intel AES-NI will further enhance performance and security by moving some of the encryption/decryption execution steps to the hardware.

5.2. Enterprise Applications

Most enterprise applications offer some kind of option to use encryption to secure information. It is a common option used for e-mail, and for collaborative and portal applications. ERP and CRM applications also offer encryption in their architectures with a database backend. Examples of databases that provide encryption options are: Oracle Database*, IBM DB2, Microsoft SQL Server*, Microsoft Access, and MySQL*. Database encryption offers granularity and flexibility at the data cell level, column level, file system level, tablespace and database level. Transparent data encryption (TDE) is a feature on some databases (Oracle Database 10gR2 and 11g[8] and Microsoft SQL Server 2008[9]) that automatically encrypts the data when it is stored to the disk and decrypts it when it is read back into memory. Retailers can use features like TDE to help address PCI-DSS requirements. University and healthcare organizations can use it to automatically encrypt their data to safeguard social security numbers and other sensitive information on disk drives and backup media from unauthorized access. Since AES is a supported algorithm in most enterprise application encryption schemes, the use of Intel AES-NI provides an excellent opportunity to speed-up these applications and enhance security (see Figure 5).

5.3. Full Disk Encryption (FDE)

FDE as shown in Figure 6 uses disk encryption software which encrypts every bit of data that goes on a disk or disk volume. While the term FDE is often used to signify that everything on a disk is encrypted—including the programs that boot OS partitions—the master boot record (MBR) is not and thus this small part of the disk remains unencrypted. FDE can be implemented either through disk encryption software or an encrypted hard drive.

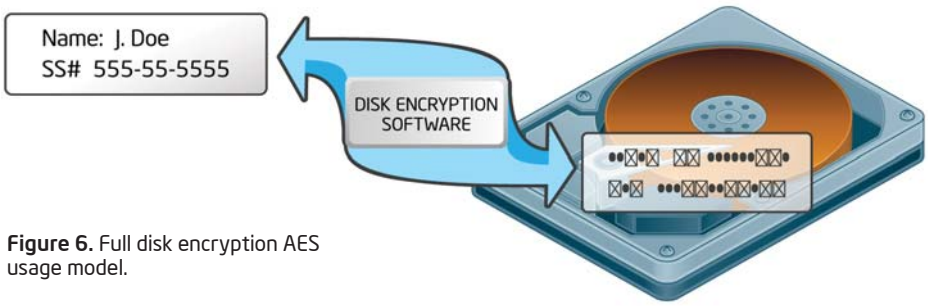

Figure 6. Full disk encryption AES usage model.

DAS is commonly connected to one or more Serial Attached SCSI (SAS) or SATA hard drives in the server enclosure. Since there are relatively few hard disks and interconnects, the effective bandwidth is relatively low. This generally makes it reasonable for a host processor to encrypt the data in software at a rate compatible with the direct-attached storage (DAS) bandwidth requirements.

Full disk encryption is growing as a security measure. Products like Windows BitLocker Drive Encryption in Windows* Server 2008, PGPdisk, and McAfee* Total Protection for Endpoint provide encryption for data-at-rest on hard disks. In addition to protecting data from loss and theft, full disk encryption facilitates decommissioning and repair. For example, if a damaged hard drive has unencrypted confidential information on it, sending it out for warranty repair could potentially expose its data. Consider, for instance, the experience of the National Archive and Records Administration (NARA). When a hard drive with the personal information of around 76 million servicemen malfunctioned, NARA sent it back to its IT contractor for repairs. By failing to wipe the drive before sending it out, NARA arguably created the biggest government data breach ever.[10] Similarly, as a specific hard drive gets decommissioned at the end of its life or re-provisioned for a new use, encryption can spare the need for special steps to protect any confidential data. In a data center with thousands of disks, improving the ease of repair, decommissioning, and re-provisioning can save money.

6. Performance Implications

This section discusses the potential performance improvements of the Intel Xeon processor 5600 series with Intel AES-NI that will help encourage greater adoption of AES encryption in many of today's usage models.

Performance can be measured by a variety of metrics and can concentrate on just a specific algorithm or a focused operation (kernel), the overall application or transaction, or the strength of the application's security. A performance study isolating kernel acceleration could result in speedup figures many times faster than a study of application acceleration, which will probably see less gain due to the effects of Amdahl's law. (This law observes how the speedup of a program using multiple processors in parallel computing is limited by the time needed for the sequential fraction of the program.)

In the marketplace today, ease of use, capability, and simply the fact that the data is encrypted to some degree, often outweighs consideration of how secure the encryption actually is.[11] Performance enhancements like those offered by Intel AES-NI could make more secure forms of encryption that involve higher computation overhead and other tradeoffs more readily adopted, making personal and corporate data in transit and storage safer in the long run.

Performance discussions in this paper are limited to results using early reference platforms based on the Intel Xeon processor 5600 series and software optimized for Intel AES-NI.

6.1. Secure Transactions Performance

If you recall from Figure 3 depicting how an SSL transaction works, in a Web server setup, the client opens a dynamic HTTPS page and downloads a static file from the server. The transaction first starts with an RSA handshake between the client and server to establish a connection, and then is followed by the SSL handshake and the bulk exchange of data which can use AES. A similar set of steps happens when the newer TLS protocol is needed.

It is expected that when encryption is turned on, performance will be degraded. However, Intel internal analysis with a Web banking workload running PHP and Windows Server 2008 R2 (Figure 7) can support more banking users on servers based on Intel Xeon processor X5600 series (formerly codenamed Westmere-EP). The servers were equipped with 48 GB RAM, 24 SSD RAID 0 arrays, and the TLS_RSA_with_AES_128_CBC_SHA cipher suite. The study has shown 23 percent more users can be supported on an Intel Xeon processor X5680 turning on SSL (encryption) compared with an Intel Xeon processor X5570 (formerly codenamed Nehalem) without SSL. Compared to a 3.0 GHz Intel Xeon processor 5160 series (formerly codenamed Wooodcrest), the number of users supported can be 4.5x more with an Intel Xeon processor X5680 series, hence supporting consolidation refresh opportunities despite having encryption enabled.

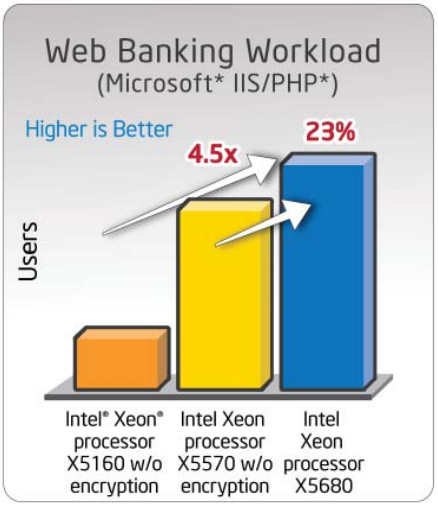

Figure 7. Intel measurement performed with Windows* Server 2008 R2 x64 Ent. Svr. PHP banking sessions/users measured with a 3.33 GHz Intel® Xeon® processor X5680 vs. a 3.0 GHz Intel Xeon processor 5160 series and a 2.93 GHz Intel Xeon processor X5570, 24 SSD RAID 0 arrays, TLS_RSA_with_AES_128_CBC_SHA cipher suite.

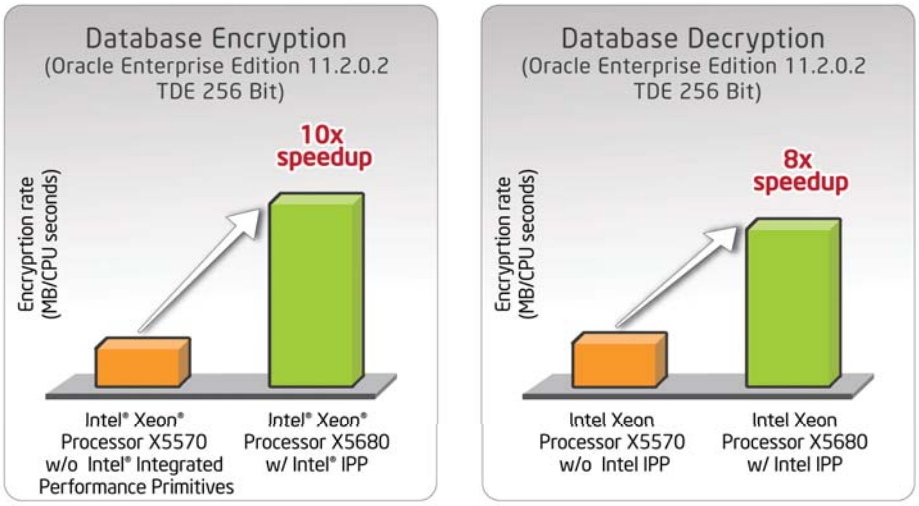

Figure 8. Intel® Xeon® processor 5600 series encryption/decryption performance with Oracle Enterprise Edition 11.2.0.2 Advanced Security TDE 256 bit.

6.2. Application-level Encryption Performance

Encryption in databases, ERP/CRM, application servers, middleware, mail servers, and hypervisors incur additional processor utilization and threading/synchronization overhead. Transparent Data Encryption (TDE) for databases incur as much as 28 percent overhead during high processor usage.[12]

In a database, decryption (reads from a drive) is more common than encryption (writes to a drive). CBC is a common mode of AES and is used in TDE.[13] CBC's main drawback is that encryption is serial (not parallelizable), hence a one-bit change in a plaintext affects all the following ciphertext blocks. CBC decryption, however, is potentially parallelizable where plaintext can be recovered from just two adjacent blocks of ciphertext.[14]

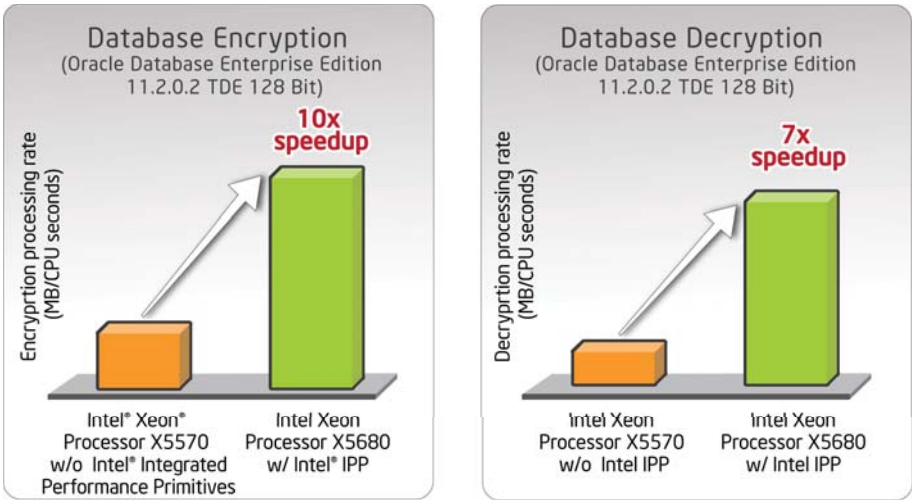

Figure 9. Intel® Xeon® processor 5600 series encryption/decryption performance with Oracle Enterprise Edition 11.2.0.2 Advanced Security TDE 128 bit.

Oracle Database Enterprise Edition 11.2.0.2 Advanced Security can now use Intel AES-NI to speed up TDE performance. As depicted in Figure 8, testing with Oracle Database Enterprise Edition 11.2.0.2 with TDE AES-256 shows 10x speedup when inserting 1 million rows 30 times into an empty table on Intel® Xeon® processor X5680 (based on Intel® microarchitecture codename Westmere, 3.33 GHz, 36MB RAM) optimized with Intel® Integrated Performance Primitives crypto library (Intel® IPP) vs. Intel® Xeon® processor X5570 (based in Intel® microarchitecture codename Nehalem, 2.93 GHz, 36MB RAM) without Intel IPP. The testing also demonstrated an 8x speedup to decrypt a 5.1 million row table. Time measured is per 8KB of data and shown as encryption/decryption processing rate in MB/CPU second. For TDE AES-128 encryption and decryption shown in Figure 9, a speedup of 10x and 7x are observed respectively. In summary, Intel Xeon processor 5600 series allow businesses to meet their compliance needs with Oracle's Advanced Security while maintaining the highest level of performance by dramatically lowering the processing cost of encryption.

6.3. Full Disk Encryption

In general, individual server drives tend to be smaller in capacity and have better latency than client drives. The first time provisioning the drive (encrypting all the data on the drive) is time consuming especially with large drives, and it could take hours. Intel Xeon processor X5600 with Intel AES-NI is expected to perform this encrypting task faster than equivalent previous generation processors. Intel internal measurement with McAfee Endpoint Encryption* for PCs (EEPC) 6.0 package with McAfee ePolicy Orchestrator* (ePO) 4.5 encrypting a 32GB Intel® X25-E Solid State Drive on a 3.33 GHz Intel Xeon

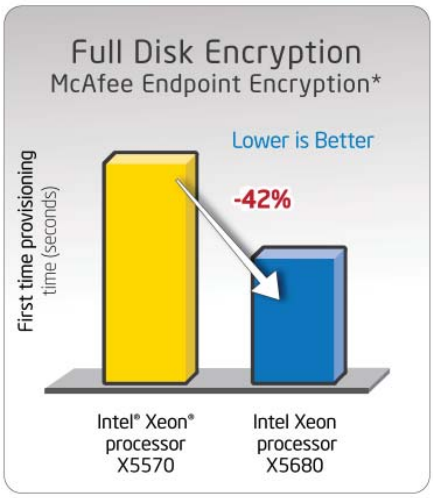

Figure 10. McAfee Endpoint Encryption* for PCs (EEPC) 6.0 package with McAfee ePolicy Orchestrator* (ePO) 4.5 encrypting a 32GB X25E SSD with a 3.33 GHz Intel Xeon processor X5680 vs. a 2.93 GHz Intel Xeon processor X5570 (server has 24GB of memory).

X5680 processor versus a 2.93 GHz Intel Xeon processor X5570 shows a 42 percent faster server SSD provisioning time (Figure 10).

7. Implementing in Applications

ISVs have three ways they can implement Intel AES-NI. They can:

- Use the instructions by using OS libraries
- Use the instructions using third-party libraries
- Code the application insertion themselves using the new instructions

Most software developers use OS crypto services or libraries for the actual encryption. Popular compilers are also enabled to support Intel AES-NI developers that write their encryption code directly. Intel has worked with OS, library and compiler vendors to optimize their software with Intel AES-NI.

7.1. Operating Systems

ISVs can take advantage of the crypto application programming interfaces (APIs) specific for a particular OS dynamically, without doing specific encryption algorithm optimization work. At the middleware and infrastructure level, the standard OS cryptographic libraries can be called from the code path.

Microsoft Crypto Next Generation (CNG) library which is available in Windows Server 2008 Release 2 and Windows 7, supports many algorithms including all key sizes of AES. The Linux crypto API, initially developed to support the cryptographic part of IPsec in the kernel, supports other uses with potential applications including encrypted files, encrypted filesystems, strong filesystem integrity, the random character device, network filesystem security, and other kernel networking services requiring cryptography. A patch with Intel AES-NI optimization has been integrated into Linux crypto-asynchronous API.

Solaris* 10 has the kernel crypto framework (kCF) which offers crypto API for other kernel modules or drivers.

7.2. Libraries

For applications that use third party libraries (Table 1) for encryption support, ISVs build their application by linking statically or dynamically to cryptographic libraries, some of which are open source. These libraries will replace the basic AES routines with optimized algorithms.

7.2.1. Intel® Integrated Performance Primitives Library

The Intel IPP cryptography function domain is a suite of pre-built public-key, symmetric and hashing functions that conform to the U.S. government's NIST Federal Information Processing Standards (FIPS) specifications. ISVs can use Intel IPP to quickly build robust, high-performance cryptographic modules and applications.

7.2.2. Java* Cryptography Extensions (JCE)

The Java Cryptography Extension (JCE) provides a framework and implementations for encryption, key generation and key agreement, and MAC algorithms. Its encryption support includes symmetric, asymmetric, block, and stream ciphers, in addition to supporting secure streams and sealed objects.

Table 1. Intel® AES-NI Optimized Libraries

LIBRARY NAME	LOCATION	INTEL® AES-NI STATUS
Intel® IPP Crypto Library	V6.1: http://software.intel.com/en-us/intel-ipp/	Available now
OpenSSL*/OpenSSH/libNSS	http://rt.openssl.org/Ticket/Display.html?id=2067&user=guest&pass=guest http://www.mozilla.org/projects/security/pki/nss/nss-3.12.3/nss-3.12.3-release-notes.html	Available now
Microsoft Cryptographic Next Generation Library	http://www.microsoft.com/downloads/en/details.aspx?FamilyId=1EF399E9-B018-49DB-A98B-0CED7CB8FF6F&displaylang=en	Available now
RSA* BSAFE*	http://www.rsa.com/node.aspx?id=1204	Q3/Q4 2010
Java* Cryptography Extensions	http://java.sun.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html	TBD
crypto++	http://www.cryptopp.com	TBD

Table 2. Compilers That Support Intel® AES-NI

COMPILER NAME	DESCRIPTION	INTEL® AES-NI STATUS
Microsoft	Visual C++* 2008 SP1: http://www.microsoft.com/downloads/details.aspx?familyid=A5C84275-3B97-4AB7-A40D-3802B2AF5FC2&displaylang=en	Available now
Intel	V11.0: http://software.intel.com/en-us/articles/intel-c-compiler-for-windows-support-resources/	Available now
GCC	4.4.0+ and Linux* Binutils 2.18.50.0.6: http://gcc.gnu.org/gcc-4.4/	Available now

JCE is designed so other qualified cryptography libraries can function as service providers and add new algorithms. Qualified providers are signed by a trusted entity.

JCE (J2SE 5.0 release) is the default crypto provider for Java applications requiring encryption. On the Solaris platform, JCE plugs into the Solaris Cryptographic Framework. It can take advantage of any mechanism available in the framework.

7.2.3. RSA* BSAFE*

RSA* BSAFE* is a very pervasive and popular free download module. RSA BSAFE supports the FIPS approved DSA, rDSA (RSA ANSI X9.31), DES and TDES Modes, and SHA-1 algorithms.

7.2.4. Crypto++

Crypto++ Library is a free C++ class library of cryptographic schemes consisting of implementations of AES, Diffie-Hellman Key Exchange, RSA cryptography, elliptic curve cryptography, and digital signature algorithm. AES-GCM, AES-CCM, and AES-CBC modules are available for download.

7.2.5. OpenSSL*

This collaborative effort provides a robust, commercial-grade, full-featured and open source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols. The OpenSSL Project also provides a strong general-purpose cryptography library. Managed by a worldwide community of volunteers, the project uses the Web to communicate, plan, and develop its OpenSSL toolkit and related documentation.

OpenSSL is based on the SSLeay library developed by Eric A. Young and Tim J. Hudson. The OpenSSL toolkit is licensed under an Apache-style license. ISVs can get and use it freely for commercial and non-commercial purposes (subject to simple license conditions).

OpenSSL is one of the few open source programs to be validated under the FIPS 140-2 computer security standard by NIST's Cryptographic Module Validation Program.

7.2.6. Linux* Kernel

The Linux kernel is released under the GNU General Public License version 2 (GPLv2) and is developed by contributors worldwide. There are proprietary licenses for some controversial binary large objects (BLOBs).

7.3. Compilers

Applications that do not leverage libraries for AES ciphers should consider adding Intel AES-NI support using an upcoming compiler (Table 2). Popular industry compilers support Intel AES-NI programming either using intrinsics or assembly.

There is an emulator available to allow early software development with Intel AES-NI. The emulator is available at the following link: http://software.intel.com/en-us/articles/pre-release-license-agreement-for-intel-software-development-emulator-accept-end-user-license-agreement-and-download.

8. Conclusion

Intel AES-NI provides a new set of processor instructions that will first be available in servers based on the Intel Xeon processor 5600 series. These instructions enable fast and secure data encryption and decryption. Since AES is the dominant block cipher, and it is deployed in various protocols, the new instructions will be valuable for a wide range of applications.

The architecture consists of six instructions that offer hardware support for AES. Four instructions support AES encryption and decryption, and the other two instructions support AES key expansion. A seventh new instruction, CLMUL, accelerates the GCM mode for AES, assisting in ECC, general purpose CRCs, as well as data de-duplication. Together, these instructions offer a significant increase in performance compared to pure software implementations.

The AES instructions have the flexibility to support all three standard AES key lengths, all standard modes of operation, and even some nonstandard or future variants.

Beyond improving performance, the AES instructions provide important security benefits. Since the instructions run in data-independent time and do not use lookup tables, they help in eliminating the major timing and cache-based attacks that threaten table-based software implementations of AES. In addition, these instructions make AES simple to implement, with reduced code size. This helps reduce the risk of inadvertent introduction of security flaws.

ISVs can employ OS crypto services and third party library optimizations to easily and efficiently integrate Intel AES-NI-optimized routines statically or dynamically. By implementing Intel AES-NI in applications, ISVs will be giving end users of the latest Intel platforms (starting with the Intel Xeon processor 5600 series) all the advantages of AES encryption, but with the much needed instruction-based acceleration that takes performance issues largely out of the equation.

This paper describes the advantages of Intel AES-NI implementation in use cases in secure commerce, enterprise applications, storage, full disk encryption, application-level encryption (e-mail, databases, etc.), and secure virtual machine migration. Due to the processor-intensive nature of encryption/decryption, Intel AES-NI becomes essential in accelerating sub-steps of the AES algorithm, allowing applications to run faster and more secure. Performance has shown that new Intel Xeon processors can run Web servers up to 23 percent faster with encryption than previous generations without encryption. These performance gains will allow better protection of data center information assets by enabling encryption in places previously not feasible for performance reasons.

For more information on Intel® AES-NI, visit www.intel.com

[1] Privacy Rights Clearinghouse, "Chronology of Data Breaches." http://www.privacyrights.org/data-breach.

[2] Charles J. Kolodgy, "IDC Encryption Usage Survey," IDC #213646, Volume: 1, August 2008.

[3] Charles J. Kolodgy, "IDC Encryption Usage Survey," IDC #213646, Volume: 1, August 2008.

[4] "States Move to Mandate Encryption of Sensitive Personal Data," The Last Watchdog on Internet Security, March 2, 2009. http://lastwatchdog.com/states-moving-mandate-encryption-sensitive-personal/.

[5] Federal Information Processing Standards Publication 197. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[6] Shay Gueron, "Advanced Encryption Standard (AES) Instructions Set - Rev 3," Intel white paper, June 2009. http://software.intel.com/en-us/articles/advanced-encryption-standard-aes-instructions-set/

[7] Six-page letter to Google's CEO, Eric Schmidt, signed by 38 researchers and academics in the fields of computer science, information security and privacy law. Together, they ask Google to honor the important privacy promises it has made to its customers and protect users' communications from theft and snooping by enabling industry standard transport encryption technology (HTTPS) for Google Mail, Docs, and Calendar. http://files.cloudprivacy.net/google-letter-final.pdf

[8] Paul Needham, "Oracle Advanced Security Data Sheet," June 2007. http://www.docstoc.com/docs/2659717/Oracle-Advanced-Security

[9] Sung Hsueh, Database Encryption in SQL Server 2008 Enterprise Edition, Microsoft SQL Server technical article, February, 2008. http://msdn.microsoft.com/en-us/library/cc278098.aspx

[10] "The Year of the Mega Data Breach," Forbes, November 24, 2009.

[11] Charles J. Kolodgy, "IDC Encryption Usage Survey," IDC #213646, Volume: 1, August 2008.

[12] Sung Hsueh, Database Encryption in SQL Server 2008 Enterprise Edition, Microsoft SQL Server technical article, February, 2008. http://msdn.microsoft.com/en-us/library/cc278098.aspx

[13] An Oracle white paper, "Oracle Database 11g: Cost-Effective Solutions for Security and Compliance," June 2009.

[14] http://en.wikipedia.org/wiki/Cipher_Block_Chaining#Cipher-block_chaining_.28CBC.29

Printed in USA 0310/JG/EMC/PDF ♻ Please Recycle 323587-001US