SIIM2013

ANNUAL MEETING
JUNE 6-9 Grapevine-Dallas, TX

ATTENDEE EXHIBITOR

General Information | Education Program | Scientific Program | Integrated Learning | Exhibit Hall | Networking & Events | Attendee Registration | Housing & Travel | Media Center

Scientific Abstract - Scientific Posters & Demonstrations

An Efficient Encryption Framework for Medical Images

Authors:

James F. Philbin, PhD, *Johns Hopkins Medical Institutions*; **Matthew Green, PhD; Yu Ning, MS; Mohmoud Ismail, MS; Michael Rushanan, MS**

Hypothesis:

Using modern encryption techniques, it is possible to store studies in an encrypted format, then transmit them in the clear and decrypt them on the receiving client. This can be done with low cost using modern algorithms and new x86 instructions.

Introduction:

As HIPAA [1] laws and patient privacy are increasingly enforced with significant financial penalties, it is becoming more important for medical imaging data to be encrypted both "at rest", i.e. in the storage system and "in flight", i.e. while being transmitted over a network. One simple way to satisfy these two constraints is to use encrypted disks for "at rest" security and TLS [2] for "in flight" security. However, encrypted disks are more expensive than regular disks and the overhead for encrypting all transmissions using TLS can be significant [3]. A better approach might be to store the medical imaging data in an encrypted format that is both secure and tamper resistant and then transmit it in the clear and decrypt it on the receiver. In this work, the Advanced Encryption Standard (AES) [5] is used in Galois/Counter Mode [6] (GCM) to encrypt medical images. AES-GCM is known to be very efficient. When combined with HMAC-SHA256 secure keys, it is also known to be very strong. AES-GCM has the additional advantage that it can create a Message Authentication Code (MAC) at the same time that it encrypts the data. Finally, AES-GCM allows the message to contain Additional Authenticated Data (AAD), e.g. header information that is transmitted in the clear, and is authenticated by the MAC.

Methods:

An efficient framework for encryption is developed to encrypt and decrypt the study data. The message format is shown in Figure 1. The format is comprised of four parts: 1) the 244-byte AAD, which contains header information, 2) the Initialization Vector that is a 12-byte random number, which is used in the encryption and decryption process, 3) the encrypted DICOM data, and 4) the 16-byte MAC that is used to authenticate the entire message including AAD, IV, encrypted data and MAC. AES-GCM has four further properties that improve efficiency: 1) the encrypted data is the same length as the original data, 2) the data can be encrypted "in place", which means that the same buffer that holds the data can be used to store the encrypted version of the data; thus, no data copying is needed; 3) GCM, as a mode of operation for block ciphers like AES, is highly parallelizable [6]; and 4) the algorithm can take full advantage of the new AES instruction set in recent x86 processors, which provides hardware instructions that implement fundamental operations for both AES [7] and GCM [8]. The MSD Header is encapsulated in the AAD. It contains 1) a UUID that is the name of the stored object, 2) the date and time it was created in UTC, 3) a flag describing whether the contained object is compressed, and 4) a 32-bit integer which describes the length of the uncompressed DICOM object. This format can be used for both MSD objects and traditional DICOM objects.

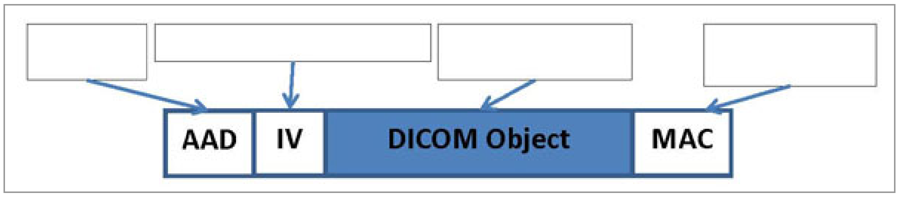

Figure 1: The format of encrypted and decrypted DICOM data

In the experiments a set of 46 DICOM studies in the enhanced Multi-Series DICOM (MSD) format [10] were used. The average study size of this dataset was 235 MB, whereas at a typical hospital, this number would usually be less than 50 MB, in one author's experience. The experiments encode each study, then encrypt it, and finally write it to disk. After that, the study is read from disk, decrypted and then parsed. The times for each phase were recorded. Two encryption methods were evaluated: 1) encrypting both metadata and bulk data, and 2) encrypting metadata only. This is possible since the MSD format separates metadata from bulk data.

Two different cryptographic libraries were used: 1) Bouncy Castle [11], a well-known open source library written in Java, and 2) Intel Integrated Performance Primitives (IPP) [12], a commercial library written in C available from Intel. All experiments were performed on a machine with an Intel® Xeon® E5-2687W CPU running at 3.10 GHz, with 16 GB of memory and a 386 GB SSD.

Results:

As can be seen from Tables 1 and 2, encrypting or decrypting an entire study is relatively fast. For the Bouncy Castle library the average time to encrypt or decrypt an entire study was 5217 milliseconds. In contrast, the IPP library was more than an order of magnitude faster, taking only 256 milliseconds on average to encrypt or decrypt the entire study. This is because it can take advantage of the AES-NI instructions as well as using multiple cores. Note that the average study size for our dataset was five times as large as that at a typical hospital, which means that the time to encrypt/decrypt a normal study would be about 1 second for Bouncy Castle and 50 milliseconds for IPP. When only the metadata is being encrypted and decrypted, the average time to encrypt/decrypt the selected studies was approximately 12 milliseconds for Bouncy Castle, with IPP again being more than an order of magnitude faster, taking on average less than 1 millisecond (Table 2).

Write Study Averages, Encrypting Metadata and Bulk Data								
	Size (KB)	**Encode (ms)**	**Encrypt (ms)**	**Write (ms)**	**Total (ms)**	**Speed-up**	**Speed (MB/s)**	**Encrypt %**
BC	240,446	16.3	5,217.0	234.2	5,467.6	-	41.0	95%
IPP	240,446	12.4	257.4	131.4	401.2	13.6	558.1	64%
Write Study Averages, Encrypting Metadata Only								
BC	240,446	16.3	11.9	234.2	262.4	-	853.5	5%
IPP	240,446	12.4	0.8	131.4	144.7	1.8	1,547.9	1%

Table 1: Write Study with Encryption

BC = Bouncy Castle, IPP = Intel Integrated Performance Primitives

*Speedup = Total*BC / *Total*IPP, *Encrypt % = Encrypt* / *Total* × 100%

Read Study Averages, Decrypting Metadata and Bulk Data								
	Size (KB)	**Read (ms)**	**Decrypt (ms)**	**Parse (ms)**	**Total (ms)**	**Speed-up**	**Speed (MB/s)**	**Encrypt %**
BC	240,446	90.0	5,198.1	13.6	5,301.7	-	42.2	98%
IPP	240,446	49.4	255.9	14.7	320.0	16.6	699.8	80%
Read Study Averages, Decrypting Metadata Only								
BC	240,446	90.0	10.8	13.6	114.4	-	1957.3	9%
IPP	240,446	49.4	0.6	14.7	64.7	1.8	3,459.4	1%

Table 2: Read Study with Decryption

BC = Bouncy Castle, IPP = Intel Integrated Performance Primitives

*Speedup = Total*BC / *Total*IPP, *Encrypt % = Encrypt* / *Total* × 100%

Discussion:

Since the MSD format separates the metadata from the bulk data into two distinct objects, it offers more flexibility because it allows different treatment of the metadata and bulk data. For example, the metadata can be both compressed and encrypted; whereas, the bulk data may already be compressed and only needs to be encrypted. Furthermore, when the bulk data contains no HIPAA covered data it does not need to be encrypted. Thus, the overhead of encrypting an MSD study is relatively inexpensive when compared to the cost of encoding and storing the study. The traditional DICOM format cannot avoid encrypting the bulk data because it is contained in the same object as the HIPAA covered metadata. It should also be noted that the encryption overhead is typically less important than the decryption overhead, because the study only needs to be encrypted once, but is usually decrypted many times for viewing.

Conclusion:

In summary, using AES-GCM with HMAC-SHA256 secure keys provides a very efficient, strong encryption foundation for medical images. The developed framework provides encryption both in the storage system and during transmission. The AES-GCM encryption also can be used to validate both file integrity in the storage system and transmission integrity over the network. Finally, the IPP library is more than an order of magnitude faster than Bouncy Castle because it takes full advantage of the new x86 instructions and multiple cores and Bouncy Castle does not.

References:

1. The Health Insurance Portability and Accountability Act. http://www.hhs.gov/ocr/privacy/hipaa/understanding/summary/privacysummary.pdf. Accessed January 15, 2013.
2. The Transport Layer Security (TLS) Protocol Version 1.2. http://tools.ietf.org/html/rfc5246. Accessed January 31, 2013.
3. Charles Shen, Erich Nahum, Henning Schulzrinne and Charles Wright, “The Impact of TLS on SIP Server Performance”, In Principles, Systems and Applications of IP Telecommunications, pp. 59 - 70. ACM 2010.
4. DICOM (Digital Imaging and Communication in Medicine), Supplement 161: Web Access to DICOM Persistent Objects by RESTful Services (WADO-RS), NEMA, Rosslyn, VA 2011. ftp://medical.nema.org/medical/dicom/supps/Drafts/sup161_13.2.pdf. Accessed January 30, 2013.
5. Federal Information Processing Standard (FIPS) for the Advanced Encryption Standard, FIPS-197. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. Accessed January 29, 2013.
6. Galois/Counter Mode – Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Galois/Counter_Mode. Accessed January 31, 2013.
7. Intel® Advanced Encryption Standard (AES) Instructions Set – Rev 3.01. http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-aes-instructions-set. Accessed January 31, 2013.
8. Intel® Carry-Less Multiplication Instruction and its Usage for Computing the GCM Mode – Rev 2.01. http://software.intel.com/en-us/articles/intel-carry-

less-multiplication-instruction-and-its-usage-for-computing-the-gcm-mode. Accessed January 31, 2013.

9. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC NIST Special Publication 800-38D. http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf. Accessed January 31, 2013.
10. Mahmoud Ismail and James F Philbin, “Fast, Storage Efficient De-identification of Medical Studies”, The DICOM International Conference and Seminar, March-2013. To appear.
11. Legion of the Bouncy Castle. http://www.bouncycastle.org. Accessed January 31, 2013.
12. Intel® Integrated Performance Primitives (Intel® IPP) 7.1. http://software.intel.com/en-us/intel-ipp. Accessed January 31, 2013.

Keywords:

- Encryption
- Multi-series
- DICOM
- Authentication
- AES-NI

ATTENDEE
- General Information
- Education Program
- Scientific Program
- Integrated Learning
- Exhibit Hall
- Networking & Events
- Attendee Registration
- Housing & Travel
- Media Center

EXHIBITOR
- Why Exhibit?
- Booth Sign Up
- Education Program Opportunities
- Sponsorship & Advertising
- Innovators & Entrepreneurs
- Already an Exhibitor?
- Housing, Travel, & Registration
- Media Center

SIIM
- About SIIM
- Membership
- Knowledge Center
- Learning Center

SOCIAL MEDIA
- Twitter
- Facebook
- LinkedIn
- YouTube
- Google +
- SIIMshare blog