Proposal for DICOM Multiframe Medical Image Integrity and Authenticity

Luiz O. M. Kobayashi,^{1,2} and Sergio S. Furuie¹

This paper presents a novel algorithm to successfully achieve viable integrity and authenticity addition and verification of n-frame DICOM medical images using cryptographic mechanisms. The aim of this work is the enhancement of DICOM security measures, especially for multiframe images. Current approaches have limitations that should be properly addressed for improved security. The algorithm proposed in this work uses data encryption to provide integrity and authenticity, along with digital signature. Relevant header data and digital signature are used as inputs to cipher the image. Therefore, one can only retrieve the original data if and only if the images and the inputs are correct. The encryption process itself is a cascading scheme, where a frame is ciphered with data related to the previous frames, generating also additional data on image integrity and authenticity. Decryption is similar to encryption, featuring also the standard security verification of the image. The implementation was done in *JAVA*, and a performance evaluation was carried out comparing the speed of the algorithm with other existing approaches. The evaluation showed a good performance of the algorithm, which is an encouraging result to use it in a real environment.

KEY WORDS: Security, image processing, integrity, authenticity, DICOM

INTRODUCTION

Security of patient data has always been a concern since the Oath of Hippocrates.¹ Among those data, medical images play a crucial role, being critical data for diagnosis, treatment, and research. Therefore, it is vital that they are kept safe from damage, be it involuntary or not. The data seen and analyzed by the physician and/or the specialist must be trustworthy.

Trustworthiness of the images can be mapped into two security services: “integrity,” allowing the

detection of data degradation, and “authenticity,” allowing the determination of the authorship of the data. Thus, compliance to legal and ethical regulations can be achieved.²

There has been a number of works about integrity and authenticity of medical images, and the digital imaging and communications in medicine (DICOM) standard does define specific instructions regarding security in Part 15.³ However, most of them deal only with single frame images so far and do not offer a satisfactory solution for multiframe modalities, like X-ray angiography (XA) or intravascular ultrasound (IVUS).

Moreover, few address the issue of performance, whose importance grows together with the size of the image. An IVUS sequence holding 1,000 images may be as big as a 100MB file. Securing large files in an adequate time is necessary for its viability in a real environment.

This work presents an algorithm using cryptographic mechanisms to achieve viable integrity and authenticity of n-frame medical images.

¹From the Heart Institute (InCor), Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo, São Paulo, SP, Brazil.

²From the Serviço de Informática (Instituto do Coração), HC/FMUSP, Av. Dr. Enéas de Carvalho Aguiar, no 44, 05403-000, São Paulo, SP, Brazil.

Correspondence to: Luiz O. M. Kobayashi, Serviço de Informática (Instituto do Coração), HC/FMUSP, Av. Dr. Enéas de Carvalho Aguiar, no 44, 05403-000, São Paulo, SP, Brazil; tel: +55-11-30695547; fax: +55-11-30695311; e-mail: luiz.kobayashi@incor.usp.br

Copyright © 2008 by Society for Imaging Informatics in Medicine

Online publication 12 February 2008

doi: 10.1007/s10278-008-9103-6

EXISTING PROPOSALS

Digital Signatures

Integrity allows detection (and in some cases, even prevention or correction) of adulteration, and authenticity makes possible to trace the author of the data, introducing the concept of responsibility. Both are usually implemented as “digital signatures.”

Digital signatures are digital counterparts of physical signatures. They are bits of information generated in two steps, “hashing” and “encrypting.” The data to be signed is used as an input to a “hash function.” This function produces a “hash,” which is a digital fingerprint of the data serving as an evidence of data integrity.

This fingerprint is, in turn, encrypted with the private key of the signer, generating the digital signature (Fig. 1). It can be assumed that it is possible to authenticate the data using a pair of public and private keys, as long as the private key is securely kept by the signer, ensuring that only he has proper access to it.⁴ Otherwise, a malicious third party could impersonate the signer, and the consequences could be catastrophic.

On the other hand, the public key can, as the name suggests, be made available to anyone who

wants to check whether the data has not been corrupted. Usually, the public key is available in “certificates,” which are digital pieces of information that allow the determination of the provenance of the key, certifying that the public key was indeed issued by the entity it should represent and not by someone else.

The verification process involves decrypting the digital signature with the public key of the signer, retrieving the hash from the signature. This hash is compared to the recalculated hash of the data. If both hash data are equal, then the signature is valid and the data can be assumed to be authentic and without degradation.

A relevant aspect is where to store the digital signature. There are two major approaches, one using headers and one using watermarking, which will be discussed in the next subsections.

The Use of Headers

The best-known proposal is DICOM Part 15. In this approach, the pixel data are digitally signed, and the signature is stored in the DICOM header according to the norms of the standard. It is important to notice that not all commercial implementations of DICOM software declare their compliance to Part 15.

Fig 1. Digital signature generation and verification processes.

There is also a proposal of creating a virtual border around the image to store the data relative to security.⁵ Strictly speaking, this is not an approach using headers, but since the data is stored outside the pixel data, it can be considered similar to the header approach.

The problem of simply using headers like in Part 15 is the weak bond between the image and the digital signature. That is, it is relatively easy to reduce image security by deleting the digital signature. A DICOM viewer compliant with Part 15 will certainly flag the image as unverified, but it will not be able to tell whether the image simply has not been signed or if the digital signature has been deleted, a tampering that has to be properly detected and identified. Therefore, it is highly desirable to be able to distinguish between unsigned images and images whose signature has been removed.

The Use of Watermarking

The process of watermarking involves embedding information in the image itself. Currently, several techniques have been developed.⁶ Embedding information about the image itself may be used as a solution to achieve image integrity and authenticity. Thus, the use of watermarking has been explored in many ways to achieve integrity and authenticity in medical imaging.^{7,8}

There are two major approaches: one involves marking the least significant bit (LSB) of the original information. Digital envelopes were proposed to be embedded to improve security,^{9,10} as well as embedding patient data themselves.^{11,12}

The other is the use of wavelets to mark the image itself¹³ or the region of interest (ROI), storing the data in a non-relevant region.¹⁴

However, watermarking by its very essence introduces image degradation. Although most algorithms make such degradation almost imperceptible to the human eye, the very idea of degrading the original image may induce severe resistance to its adoption.

Recently, reversible watermarking has been seen as a potential way to overcome this shortcoming, where the embedded data are retrieved and the original image is restored. This is a field where several researches are being developed,^{15,16} including embedding digital signatures in volumetric images,¹⁷ an approach based on a previous work that uses lossless data compression to free up space

to store the signature.¹⁸ It should be noted that reversible watermarking introduces another step of image processing to restore the image back to its original state.

Another restriction is that almost none of the proposed algorithms in existing works on medical image watermarking offers insights about how they would work for multiframe images, the volumetric image work being an exception. It is important to point out that there is the obvious solution, where the algorithms are applied for each frame, but this does not detect the deletion of entire frames. This issue has to be adequately addressed.

Moreover, the performance is seldom discussed, which may render the algorithms unviable in a real environment. This is especially relevant for reversible watermarking, needing proper evaluation for large datasets, to ensure that the image restoration process offers competitive performance.

PROPOSED ALGORITHM

There are two major challenges that should be solved by the algorithm:

1. The algorithm must provide a solution for providing integrity and authenticity for n-frame images
2. The algorithm must offer an adequate performance. This should reflect not only in the processing itself, but also in a lesser amount of data being sent over the network. Minimization of data overhead is a relevant issue.

Therefore, a new algorithm was developed, which uses cryptographic mechanisms to add integrity and authenticity for n-frame medical images.

For adding security, the following flow is proposed (Fig. 2).

To begin with, the first frame (frame 0) is signed conventionally. Then, the resulting signature is stored in the DICOM header and used as a key to cipher the remaining frames, generating the ciphered frames C1, C2, ..., Cn and their respective message authentication code (MAC), which will be inputs to generate MMAC, the MAC of the MACs. This MMAC is in turn signed with the private key, and the signed MMAC is used as a key to encrypt the first frame, generating C0.

The final DICOM file will contain the DICOM header and the frames C0, C1, ..., Cn.

The encryption of frames 1 to n is done in a cascading process, as shown above (Fig. 3). The MAC1 generated by ciphering the first image is used as the nonce, i.e., the initialization data, to encrypt the second, and so on. This was done because the frames tend to be similar to each other. Therefore, if encrypted with exactly the same key and nonce, the ciphered frames also tend to be similar to each other, which may pose as potential security vulnerability.

In addition, if each frame is treated independently, then it will not be possible to detect the deletion of the entire frames. Thus, the dependency introduced by the cascading scheme is also a measure to avoid this weakness.

One important issue is how to choose the initial nonce. It must be something easily obtainable by the verifying party and be unique to avoid using the same nonce for different images. Therefore, given the DICOM structure, the nonce was chosen to be the hash of the SOP Instance UID. Additional data (patient information, etc.) may also be used as inputs to ensure further authentication.

Furthermore, for the verification process, the following flow is proposed (Fig. 4).

As it would be expected, the decryption process is very similar to the encryption process. First, the digital signature is retrieved from

the header. Then, the cascading scheme is used to retrieve images 1, 2, ..., n , along with their MAC, which in turn serve as inputs to generate MMAC. MMAC is decrypted by the public key present in the DICOM header and then used as the key to decipher the first frame, retrieving the frame 0.

Finally, there is the digital signature verification itself, as shown in the figure below (Fig. 5).

Therefore, the algorithm allows the detection of any tampering because the decryption process to retrieve the original frames will succeed if and only if all the data are exactly the original and unmodified data.

If a malicious party alters the digital signature or the nonce in any way, then it is trivial to notice that the decryption keys will be changed, and thus, the verification process will detect the adulteration.

A modification on any of the frames will also be detected, for each frame generates the nonce to decipher the next frame in the sequence. Changing any frame will compromise data procurement for all the subsequent frames. The only exception to this is C0, which is the last batch of information to be processed. However, any tampering with C0 will be correctly detected in the digital signature verification.

Fig 2. The proposed flow for security addition.

Fig 3. Detail of the encryption in cascade and the MMAC generation.

Moreover, if C_i is a frame, where $0 < i < n$, then a natural consequence of adulteration is the generation of visual evidence of tampering, since all frames C_j , $i < j$, will not be retrieved correctly. Therefore, an end user will have an obvious, visual proof of whether the images are trustworthy or not.

Implementation for Single Frame Images

For single frame images, the process is trivial: the image is signed and then encrypted with the digital signature and the nonce generated in the same fashion as presented before (Fig. 6).

Fig 4. The proposed flow for security verification.

Fig 5. The verification process of the digital signature, after retrieving the original data.

The verification is also trivial: first, the digital signature is retrieved from the header. Then, the image is decrypted and the digital signature checked (Figs. 7 and 8).

Implementation Details

The implementation was entirely done using *JAVA* in the Eclipse environment.

For the digital signature itself, the chosen algorithm was SHA-1 with RSA. This is the most traditional algorithm, being recognized in the DICOM standard as well.

The algorithm used to cryptograph the images was Phelix, a stream cipher with several interesting properties, including its speed and the quasi-

simultaneous generation of a message authentication code with the encrypted data.¹⁹

And the MMAC was generated using the Whirlpool algorithm. This is also a somewhat recent algorithm which generates a hash of 512 bits, being a stronger alternative to the most popular hash algorithms. It was adopted and standardized by ISO.²⁰

RESULTS

Performance Comparison

To evaluate the performance of the proposed algorithm, a batch of tests was made to offer a comparative study of performance.

Fig 6. The flow for adding security for single frame images.

Fig 7. The flow for verifying security for single frame images.

The algorithms used for comparison were:

1. The DICOM standard digital signature algorithm, as specified in Part 15. This algorithm involves simply signing all pixel data and storing the signature in the header. As this is the approach established by the standard, it is vital to see how fast the proposed algorithm when compared to it.
2. Encryption with AES in Galois Counter Mode (GCM). Advanced encryption system (AES) is now the most widespread algorithm for data encryption. To fulfill certain needs, operation modes were developed for AES, one of which is GCM. This mode emulates a stream cipher encryption and also generates an authentication code for integrity verification.²¹ Therefore, being an alternative to the proposed algorithm, it is interesting to compare one with the other.
3. Watermarking using Haar wavelets. There are two major approaches to watermarking medical images. One of them is decomposing the image using Haar wavelets and marking its coefficients in a way that the reverse Haar transformation will not degrade the data in an irreversible fashion. For this evaluation, an algorithm based on the works of Giakoumaki et al.,²² suggesting the use of multiple watermarking, was implemented. The algorithm does a four-level Haar decomposition, marks the

Fig 8. Sample of images. On the left, the original images. On the right, the images after encryption.

coefficients, and performs the reverse transformation for each frame.

4. LSB watermarking. The other popular watermarking method is embedding the data in the LSB of the image. For this paper, an algorithm based on the works of Zhou and Huang⁹ was implemented, signing each frame, enveloping the signature, and embedding it in the pixel data themselves.

Two kinds of tests were done. First, all algorithms secured multiframe images to analyze their performance according to the size of the image. Obviously, the larger the data, the slower they will be, meaning that more time would be necessary to do all the processing.

The second test was with multislice images, to evaluate the speed according to the number of images. Each slice is stored in a separate file and it has a small size, but each series can have literally hundreds of slices. Again, the bigger the number of slices, the slower they will be. But “how slow” is an uncertainty that must be addressed for the multiframe and for the multislice tests.

All simulations were done in a Pentium 4 with 1 GB RAM, 3.0 GHz. All images were anonymized before the tests, and no particular information can be derived from the images, preserving patient’s privacy. The result was as follows.

As can be seen in Figures 9 and 10, for multiframe image encryption, the Haar watermarking was by far the worst, being as slow as six times when compared to the other algorithms. The wavelet transformation, marking and inverse transformation, is computationally heavy, which reflects in its poor performance both for multiframe and multislice images.

As the implementation used the most basic approach to implement Haar wavelets, optimization can be done to reduce the computational load of the process, but it is unlikely that it will be faster than LSB watermarking, whose design is simpler by its very nature.

The quickest approach was the approach of the DICOM standard. The proposed algorithm followed suit, with small differences in performance for larger files.

The AES in GCM showed a similar performance as well, although being slower in larger files than the proposed approach. LSB watermarking was faster than the implemented Haar

watermarking but still took twice the time spent by the method defined in this work.

For the decryption and verification process, the results are shown in Figures 11 and 12.

Similarly to encryption, the Haar watermarking showed the worst performance, being far slower than the other implementations. For larger images, it was more than 15 times slower than the other algorithms.

Again, the standard digital signature algorithm showed the best performance. The LSB watermarking approach was faster than the encryption algorithms for security verification, although the method proposed in this paper was almost as fast as the LSB watermarking. As LSB watermarking requires only a quick retrieval of the signature and its verification, its speed is naturally closer to the plain digital signature method.

For multislice images, the outcome was more or less similar to the multiframe test, as can be seen in Figures 13 and 14. The Haar watermarking offered once again the worst performance.

The AES in GCM encryption algorithm showed a poor performance for multislice images as well, although not as poor as the Haar watermarking. Still, it took twice the time needed by this paper’s method to apply and to verify data security. This is related to its initialization for each slice, a process which is inherently slow.

The encryption time and decryption time were approximately the same for each algorithm. Also in this test, the LSB watermarking showed a performance almost equal to the standard digital signature approach.

DISCUSSION AND CONCLUSION

The method presented in this paper offers a solution to secure n-frame medical images, with a specific approach to address the issue of multiframe images, filling in a gap left by previous works in this field.

It is interesting to point out that the proposed scheme is, by its own nature, directed to DICOM images. To deploy this in the current DICOM scenario, it has to be noticed that the current standard does not allow a digital signature representing only a partial portion of the pixel data and therefore a private tag is needed for the signature storage, along with a private SOP Class or a

Fig 9. Multiframe encryption time graph.

Fig 10. Multiframe encryption time graph, showing the four quickest algorithms' performance.

Fig 11. Multiframe decryption and verification time graph.

Fig 12. Multiframe decryption and verification time graph, showing the four quickest algorithms' performance.

private Transfer Syntax to keep interoperability with current deployments.

To incorporate the scheme in the standard, on the other hand, would be rather simple, needing only to define either a SOP Class or a Transfer Syntax and appropriate Security Profiles to explicit the use of the scheme. There would be no need to define further tags or modules to store additional data.

The proposed algorithm provides a higher security level than the standard digital signature approach as defined in DICOM Part 15, since the bond between the signature and the pixel data is stronger. It is harder to tamper with the data because its visualization goes necessarily through security verification. Furthermore, the slightest change in the signature and/or in the data is detected—depending on the adulteration, one can have an evident and visible proof of tampering, showing noise to the user instead of any meaningful data.

To study its viability in a real environment, a comparative performance evaluation was carried out with several methods proposed in existing works. As a result, the proposed algorithm showed competitive speeds, rivaling with the DICOM method albeit a little bit slower, and besting some

of the existing methods while offering a stronger security than the fastest algorithms.

The cascading approach seen in this work, however, brings along two constraints in terms of performance: the first is that to retrieve a frame j , $j > 1$, one must retrieve all frames $k < j$ before that. If $j = 1$, all frames must be deciphered first, affecting the performance of frame-based queries. The second constraint is the difficulty in parallelizing the algorithm, a technique that could improve performance and reduce the times needed for applying and verifying image security mechanisms.

Both constraints can be alleviated by splitting the multiframe image into several blocks and applying the approach for each block. In this case, there must be a way to link all blocks together to keep overall integrity and authenticity.

This method makes it virtually impossible for damage in the images to go unnoticed, except in the case where the legitimate owner of the data is the malicious party, to which there is no technical solution. Correct definition and enforcement of policies and procedures of the institution may reduce this risk.

Fig 13. Multislice encryption time graph.

Fig 14. Multislice decryption time graph.

Another matter of importance is that unlike all the previous approaches (watermarking and use of headers), the method proposed in this work allows the user to access only what has not been meddled with. This is a substantial shift of security level, which brings along a higher reliability of data but introduces also vulnerability in terms of digital evidence. It is easier to destroy inconvenient information, since all that one has to do is to change one bit of the pixel data. This risk should be taken into account when dealing with aspects like access control and audits.

This brings along an interesting discussion on whether the user should or should not be able to see tampered images. In the method presented in this work, tampered images are simply presented as noise after data decryption, even for the smallest adulteration. This means that if one is seeing any meaningful image, then one can be sure that it is exactly what should be seen.

On the other hand, it could be argued that adulteration in non-ROI portions should not prevent the user from viewing the images, as all

the essential information is present and unmodified. In this case, the security mechanism would have to be able to pinpoint the location of the tampering and then decide whether it should show the image or not. Studies must be carried out to determine the viability of such mechanism.

This paper presents a new alternative to the existing methods in medical image integrity and authenticity, aiming at a more trustworthy digital environment in the healthcare field with little overhead and a minor decrease in performance, providing an adequate cost-benefit ratio for deployment when compared to previous approaches.

To achieve satisfactory levels of security, several methods and procedures must be developed and evaluated, and this work is part of this larger effort to provide adequate security to clinical information in all levels.

ACKNOWLEDGMENT

The authors would like to thank CNPq and FINEP for financial support for this research.

REFERENCES

1. Oath of Hippocrates. In: Harvard Classics, v. 38. Boston: P.F. Collier and Son, 1910
2. IMIA Code of Ethics for Health Information Professionals, IMIA, 2002. Available: http://www.imia.org/English_code_of_ethics.html. Accessed 20 April 2007
3. Digital Imaging and Communications in Medicine (DICOM) Standard: DICOM. Available: <http://medical.nema.org/dicom/2006/>. Accessed 05 May 2007
4. Stallings W: Cryptography and Network Security – Principles and Practice, New Jersey: Prentice Hall, 1995
5. Trichili H, Bouhlef MS, Derbel N, Kamoun L: A new medical image watermarking scheme for a better teleradiology. In: 2002 IEEE International Conference on Systems, Man and Cybernetics. Session MP1B
6. Albanesi MG, Ferretti M, Guerrini FA: Taxonomy for Image Authentication Techniques and Its Application to the Current State of the Art, Proceedings on 11th International Conference on Image Analysis and Processing, pp 26–28
7. Coatrieux G, Maître H, Sankur B, Rolland Y, Collorec R: Relevance of watermarking in medical imaging, 2000 IEEE EMBS Conference on Information Technology Applications in Biomedicine, pp 250–255
8. Coatrieux G, Maître H, Sankur B: Strict integrity control of biomedical images. In: SPIE Conference 4314: Security and Watermarking of Multimedia Contents III
9. Zhou XQ, Huang HK: Authenticity and integrity of digital mammography images. IEEE Trans Med Imag 20(8): 784–791, 2001
10. Cao F, Huang HK, Zhou XQ: Medical image security in a HIPAA mandated PACS environment. Comput Med Imaging Graph 27:185–196, 2003
11. Acharya UR, Bhat PS, Kumar S, Min LC: Transmission and storage of medical images with patient information. Comput Biol Med 33:303–310, 2003
12. Miaou S, Hsu C, Tsai Y, Chao H: A secure data hiding technique with heterogeneous data-combining capability for electronic patient records. In: World Congress on Medical Physics and Biomedical Engineering, Proceedings of, Session Electronic Healthcare Records, IEEE-BEM
13. Giakoumaki A, Pavlopoulos S, Koutsouris D: A Multiple Watermarking Scheme Applied to Medical Image Management, Proceedings of the 26th Annual International Conference of the IEEE EMBS 2004, pp 3241–3244
14. Wakatani A: Digital watermarking for ROI medical images by using compressed signature image, 35th Hawaii International Conference on Systems Sciences, Proceedings of, pp 2043–2048
15. Zain JM, Baldwin LP, Clarke M: Reversible watermarking for authentication of DICOM images, Conference Proceedings of the 26th Annual International Conference on Engineering in Medicine and Biology Society (EMBC 2004), v. 2, pp. 3237–3240
16. Guo X, Zhuang T: A lossless watermarking scheme for enhancing security of medical data in PACS, Proceedings of the SPIE Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation 5033:350–359, 2003
17. Zhou Z, Huang HK, Liu BJ: Three-dimensional lossless digital signature embedding for the integrity of volumetric images, Proceedings of the SPIE Medical Imaging 2005: PACS and Imaging Informatics, v. 6145, 61450R, 2006
18. Zhou Z, Huang HK, Liu BJ: Digital Signature Embedding (DSE) for Medical Image Integrity in a Data Grid Off-Site Backup Archive, Proceedings of the SPIE Medical Imaging 2005: PACS and Imaging Informatics 5748:306–317, 2005
19. Whiting D, Schneier B, Lucks S, Muller F: Phelix – Fast Encryption and Authentication in a Single Cryptographic Primitive. Available: <http://www.schneier.com/paper-phelix.pdf>. Accessed 05 June 2007
20. ISO: ISO/IEC 10118–3:2004 Standard. Available: <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=39876&scopelist=>. Accessed 05 June 2007
21. McGrew DA, Viega J: The Galois/Counter Mode of Operation (GCM), NIST Computer Security Division, Computer Security Resource Center, 2005
22. Giakoumaki A, Pavlopoulos S, Koutsouris D: Secure and efficient health data management through multiple watermarking on medical images. Med Biol Eng Comput 44:619–631, 2006