Research Paper ■

A Cryptologic Based Trust Center for Medical Images

STEPHEN T. C. WONG, PHD

Abstract **Objective:** To investigate practical solutions that can integrate cryptographic techniques and picture archiving and communication systems (PACS) to improve the security of medical images.

Design: The PACS at the University of California San Francisco Medical Center consolidate images and associated data from various scanners into a centralized data archive and transmit them to remote display stations for review and consultation purposes. The purpose of this study is to investigate the model of a digital trust center that integrates cryptographic algorithms and protocols seamlessly into such a digital radiology environment to improve the security of medical images.

Measurements: The timing performance of encryption, decryption, and transmission of the cryptographic protocols over 81 volumetric PACS datasets has been measured. Lossless data compression is also applied before the encryption. The transmission performance is measured against three types of networks of different bandwidths: narrow-band Integrated Services Digital Network, Ethernet, and OC-3c Asynchronous Transfer Mode.

Results: The proposed digital trust center provides a cryptosystem solution to protect the confidentiality and to determine the authenticity of digital images in hospitals. The results of this study indicate that diagnostic images such as x-rays and magnetic resonance images could be routinely encrypted in PACS. However, applying encryption in teleradiology and PACS is a tradeoff between communications performance and security measures.

Conclusion: Many people are uncertain about how to integrate cryptographic algorithms coherently into existing operations of the clinical enterprise. This paper describes a centralized cryptosystem architecture to ensure image data authenticity in a digital radiology department. The system performance has been evaluated in a hospital-integrated PACS environment.

■ **JAMIA.** 1996;3:410–421.

Medical images form the cornerstone of patient records and often are at the heart of the patient's diagnosis, determination of therapy, and follow-up. They are used not only by radiologists, but also by other clinicians and specialists, such as medical oncologists, radiotherapists, surgeons, neurologists, cardiologists, dermatologists, pathologists, and primary physicians. The trend in medical imaging is increasingly toward a digital and multimedia orientation. The goals are to represent medical images in digital form supporting image transfer and archiving and to manipulate visual information for various clinical services, such as teleradiology and diagnostic workups. Another push is from the picture archiving and communication systems (PACS) community, which envisions an all-digital radiology environment in hospitals for acquisition, storage, communication, and display of large volumes of medical images.[1] The PACS technology provides a systems integration solution for these islands of automation and facilitates the extraction of the rich information contained in multimodality images. Several large-scale PACS have been successfully put into clinical operation and trials.[1] The new thrust of PACS development is to integrate complementary textual information of clinical systems into the central image archive.[2,3]

Affiliation of the author: Department of Radiology, School of Medicine, The University of California, San Francisco, CA.

Correspondence and reprints: Stephen T. C. Wong, PhD, Department of Radiology, UCSF, Box 0628, 505 Parnassus Avenue, San Francisco, CA 94143.
e-mail: stephenwong@radmac1.ucsf.edu

Received for publication: 5/8/96; accepted for publication: 7/1/96.

Along with this digital radiology world comes the problem of establishing trust in medical documents that exist only in the easily altered memory of a computer. Trust can be defined in terms of authenticity—i.e., detect unauthorized modification of image data—and confidentiality—i.e., prevent unauthorized disclosure of image data. One of the advantages of printed film or text is its authenticity—when we see ink on paper or images in print, we feel that these are immutable records, not subject to manipulation or tampering without leaving traces. Few of us, however, have this level of faith in the immutability of medical records committed to electronic media.

People who seek unauthorized access to online medical records do so for several reasons: unauthorized release, industrial espionage, sports hacking, computer theft, and vandalism, to name a few.[4] Examples of unauthorized release include failure to obtain informed consent in writing from the patient or failure to seek an approval from the responsible authority. Industrial espionage occurs, for instance, when a health insurance company seeks a business advantage by obtaining the confidential patient population demographics stored in a competitor's databases. Hacking is another form of unauthorized access by people who view such activities as a sport, and hackers usually leave the data and systems intact. In contrast, computer theft and vandalism are the most dangerous forms of unauthorized access. Individuals penetrating an online medical database seek to steal or alter information about patients and harm the medical system.

Many means have been proposed to improve the security of online medical information: limit physical access to the network, change user passwords frequently, create firewalls to isolate information from other networks, and enforce administrative procedures for data security. Cryptography is one of the strongest and most mathematically sound methods to ensure trust in computerized medical data. There are two major cryptographic techniques: key-based encryption and digital time stamping. These techniques complement one another. Key-based cryptography associates the content of an image with the originator by using one or two distinct keys and prevents unauthorized disclosure of the image.[5] Digital time stamping, on the other hand, generates a characteristic "digital fingerprint" for an image when it is first generated by using a mathematical hash function that permits detection of subsequent modifications. Digital time stamping is not a form of encryption. Research in the past two decades has concentrated on authenticating textual data. The growing use of digital medical images, however, poses new challenges due to their large size and different user requirements. This study investigates the appropriateness of various cryptographic algorithms for improving the security of medical images and derives new methods that incorporate these algorithms seamlessly into digital radiology operations, especially PACS and teleradiology.

Methods

Terminology

This section introduces the basic nomenclature. A message is called plaintext. The process of disguising a message so as to hide its content is called encryption. An encrypted message is called, ciphertext. The process of converting ciphertext back into plaintext is called decryption. Cryptography is the art and science of keeping a message secure, and cryptoanalysis is the art and science of breaking ciphertext.

A cryptographic algorithm, also known as a cipher, is the mathematical function used for encryption and decryption. To encrypt a plaintext message, apply an encryption algorithm to the plaintext. To decrypt a ciphertext message, apply a decryption algorithm to the ciphertext. If the strength of the security provided by an algorithm is based on keeping the nature of the algorithm secret, it is called restricted (e.g., Zenith's video-scrambling algorithm). By today's data security standards, restricted algorithms provide woefully inadequate security. They are easy to break by experienced cryptanalysts and are not suitable for a large or changing group of users.

Meanwhile, many users and developers have the misconception that data compression can provide protection as certain compression algorithms scramble image data into visually unrecognized forms. The truth is that data compression, similar to restricted cryptographic algorithms, provides little protection once the compression algorithm used is known by the intruder. For high-security applications, all modern encryption algorithms use a key, which can take on one of many values (larger is better). Figure 1 shows the process of encryption and decryption with keys.

A protocol is a series of well-defined steps, involving two or more parties, designed to accomplish a task. A cryptographic protocol is one that uses cryptography. A self-enforcing protocol is the best type of cryptographic protocol because it is independent of the trustworthiness of people or the secrecy of the cryptographic algorithms used. The protocol itself guarantees fairness; if one of the parties tries to cheat, the other party immediately detects the cheating and the protocol stops. Whatever the cheating party hoped would happen doesn't happen. In the teleradiology

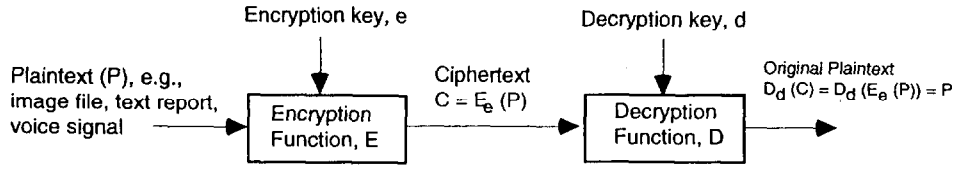

Figure 1 Encryption and decryption with keys.

and PACS environment, the system designer should strive to design and implement cryptographic protocols that are self-enforcing, cost-effective to implement, and can ensure a high level of security.

Cryptographic Algorithms

A cryptographic algorithm is the mathematical function used for encryption and decryption. Table 1 classifies key-based algorithms according to the nature of the encryption and decryption keys.

Secret Key Encryption

In a secret or private-key algorithm, the encryption key can be calculated from the decryption key, and vice versa. In many such cryptosystems, the encryption and the decryption keys are the same. These algorithms require the sender and receiver to agree on a key before they pass messages back and forth. This key must be kept secret; therefore, the level of security provided by such symmetric algorithms rests in the key. The Data Encryption Standard (DES), adopted by the federal government in 1976 and authorized for use on all unclassified government communications, is an example of a secret key algorithm.[6] The key is a 56-bit number and can be changed at any time.

Figure 2 illustrates an example of a secret key cipher on magnetic resonance images (MRIs). The cipher used is based on the International Data Encryption Algorithm (IDEA). The IDEA cipher is a newer and more secure cipher than DES.[7] Its key length is 128 bits—over twice as long as DES. Assuming that a brute-force attack is the most efficient, it would require 2^{128} (10^{38}) encryptions to recover the key. In contrast to the DES cipher, no papers have been published on breaking IDEA messages so far. In addition, there are no obvious patterns in the ciphertext. Figure 3 illustrates this with the even histogram distribution of the encrypted image slice in Figure 2. Pixel values of the 8-bit image slice range from 0 to 255.

The main drawback of secret-key algorithms is that anyone with the key can both encode and decode messages. Thus, any message can be compromised when the key is intercepted. Managing keys involved in a cryptographic protocol creates another problem. Assuming a separate key is used for each pair of users in a network, the total number of keys increases rapidly as the number of users increases. For n users, the total number of keys needed is $(n \times (n - 1))/2$; e.g., 10 users need 45 different keys to talk with one another, while 100 users need 4,950 keys. Such complexity of key management is not feasible for large user groups in a hospital or health maintenance organization (HMO) environment. Further, it is impossible to send someone a secret message unless the sender already can send the receiver a secret message—that is, the sender cannot communicate with the receiver without prior arrangement.

Public-Key Encryption

Public-key algorithms solve the secret-key management problem by having two different keys: one public and one private.[8] Information is encoded by the sender with the recipient's public key but can only be

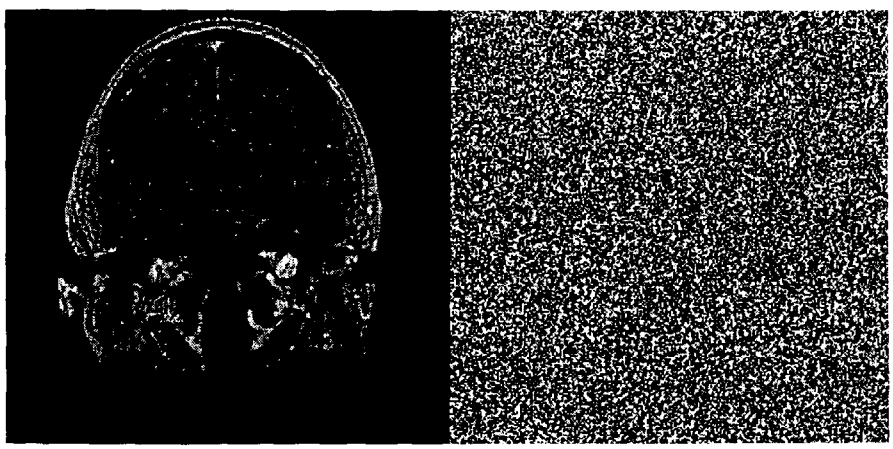

Figure 2 The original MR (left) image slice and the corresponding MR (right) image slice encrypted using the IDEA algorithm.

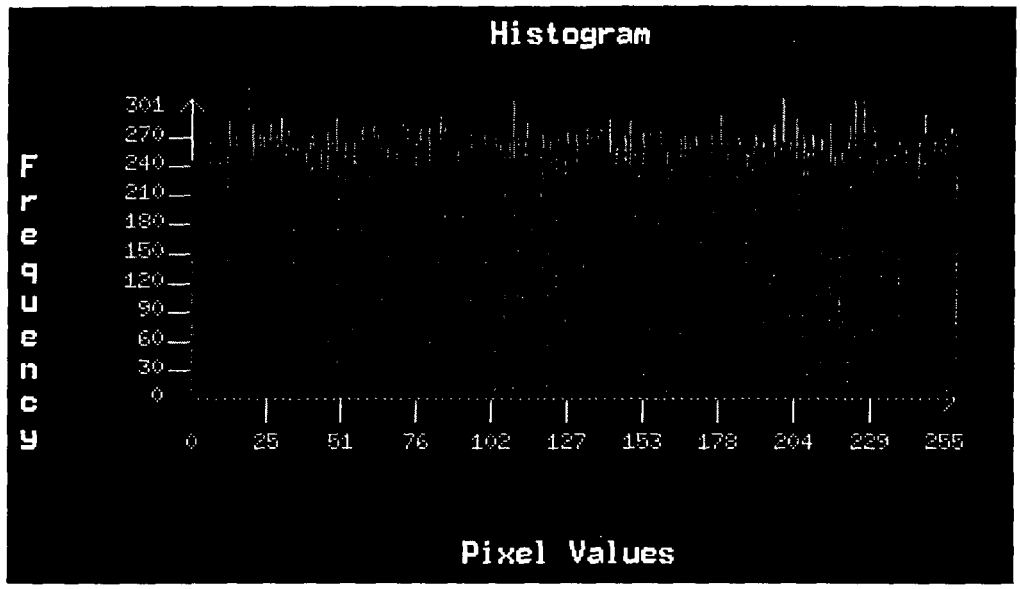

Figure 3 The histogram distribution of the encrypted MR image slice in Figure 2.

Table 1 ■

Three Types of Key-based Cryptography

Type	Nature of keys	Characteristics
Secret-key cryptography	Encryption key (e) = Decryption key (d)	e and d are private
Public-key cryptography	e ≠ d	e public, d private
Digital signature	e ≠ d	e private, d public

decoded by a recipient who possesses the private key. Moreover, the public key contains no hint as to the nature of the private key—it is computationally impossible to deduce the private key from the public key. Anyone with the public key (which, presumably, is made public by the owner) can encrypt a message but can not decrypt it. Only the person with the private key can decrypt the message.

Although public-key algorithms have better and more reliable key management over secret-key algorithms, they are much slower in execution. An example is the Rivest, Shamir, and Adleman (RSA) public-key cryptography, which derives its strength from the difficulty of factoring large numbers.[9] The public and private keys used in RSA are functions of a pair of large (100 to 200 digits or even larger) prime numbers. Recovering the plaintext from one of the keys and the ciphertext is conjectured to be equivalent to factoring the product of the two primes. It is, however, not practical to use pure RSA with large keys to encrypt and decrypt long messages. A 1,024-bit RSA key would decrypt messages about 4,000 times slower than the secret-key cipher. Furthermore, the workload to exhaust all the possible 128-bit keys in the IDEA cipher would roughly equal the factoring workload to crack a 2,304-bit RSA key, which is quite a bit bigger than the 1,024-bit RSA key size that most people use for high-security applications. Given this range of key sizes, and assuming there are no hidden weaknesses in the secret-key cipher, the weak link in this approach of using both private- and public-key cryptography is in the public-key cipher.

Public-key cryptography is attractive not because it is intrinsically stronger than a secret-key cipher—its appeal is that it helps one manage keys more conveniently. Subsequently, the use of public-key encryption for large medical images is better accomplished by using a high-quality, yet faster, single-key encryption algorithm to encipher the message. This original unenciphered message is the plaintext. In a process transparent to the user, a temporary random key, created just for this one session, is used to conventionally encipher the plaintext file. The recipient's public key is then used to encipher this temporary random conventional key. This public-key–enciphered conventional "session" key is sent along with the enciphered text (ciphertext) to the recipient. The recipient uses his or her own secret key to recover this temporary session key and then uses that key to run the fast, conventional single-key algorithm to decipher the lage ciphertext message.

Digital Signature

Digital-signature algorithms are used to prove authorship of, or at least agreement with, the contents of the computerized document. Digital signatures can be accomplished in some public-key algorithms by en-

crypting messages with the sender's private key and decrypting them with the sender's public key. Encrypting a radiologic image or report using the physician's private key generates a secure digital signature.

When the focus is on the authenticity of medical documents rather than their confidentiality, a digital signature can often be implemented with one-way hash functions.[10] A hash function takes an input data string and converts it to a fixed-size, often smaller, output data string. For a one-way hash function, it is easy to compute a hash value from an input string, but it is difficult to generate a string that hashes to a particular value. Commonly used hash functions in cryptography return values on the order of 128 bits long, so that there are 2^{128} possible hashes. The number of trials required to find a random string with the same hash value as a given string is 2^{128}, and the number of trials required to find two random data strings having the same (random) hash value is 2^{64}. The hash value, known also as the digital fingerprint or message digest (MD), is somewhat analogous to a "checksum" or CRC error checking code in that it compactly represents the message and is used to detect changes in it. Unlike a CRC, however, it is not computationally feasible for an attacker to devise a substitute message that would produce an identical message digest. With the public-key encryption, the message digest is encrypted by the secret key to form a signature. Hash functions are also used extensively in digital time-stamping methods.

Digital Time Stamping

Another proposal for certifying the contents of a document involves using a one-way hash function to create a type of digital time stamp of the document.[11,12] Many secure one-way hash functions are publicly known, however. A forger thus could alter a message, compute a new message digest, and simply attach it to the bogus message. One strategy is to encrypt the hash value with the sender's secret key and attach that value to the original message. The receiver computes a new hash value from the message and compares it with the one recovered from the sender's public key. If they match, then the message was not altered. It is worth noting that computing the hash value of a file is a much faster process than encrypting the entire file.

Another way to strengthen the credibility of a document's time stamp would be to send its hash-value fingerprint to a central time-stamp service. This service would attach the time of arrival and put both in permanent storage. Any question about a document's date and authenticity could be settled by checking the time-stamp service. This observation leads to the concept of digital trust centers for authenticating large volumes of medical images (see Digital Image Trust Centers, below).

Key Length

From the user's point of view, cryptographic keys are similar to the passwords used to operate automatic teller machines and to control access to computer systems. Just as different computer systems allow passwords of different lengths, different encryption algorithms use keys of different lengths. As with passwords, the longer the key, the stronger the security that the algorithm provides. The common way to crack a key used in a robust secret-key algorithm is by brute-force attacks (e.g., by trying every possible key, one after another, until one of the tries succeeds in decrypting the ciphertext).

It is easy to calculate the complexity of a brute-force attack. If the key is eight bits long, there are 2^8, or 256, possible keys. Therefore, it will take 256 attempts to find the correct key, with a 50% chance of finding the key after half of the attempts. If the key is 56 bits long, as in the case of DES, then there are 2^{56} possible keys. Assuming a computer program can try a million keys per second, it will take 2,285 years to find the correct key. If the key is 128 bits long, as in case of IDEA, it will take 10^{25} years. For $1 million, a brute-force cracking machine could be built to crack a 56-bit DES key in an average of 3.5 hours (results guaranteed in 7 hours).[13] This cost is within the budgets of most large companies and many criminal organizations. Fortunately, breaking an 80-bit or higher key is still extremely difficult, if not beyond the realm of possibility, at this stage.

In contrast with secret-key algorithms, breaking public-key algorithms does not involve trying every possible key. Instead, it involves trying to factor the large numbers that are the product of two large primes (see Public Key Encryption, above). Factoring large numbers is difficult. Table 2 lists public-key module lengths whose factoring difficulty roughly equals the

Table 2 ■

Private-key and Public-key Key Lengths with Similar Resistance to Brute Force Attacks

Private-key Key Length (bits)	Public-key Key Length (bits)
56	384
64	512
80	768
112	1792
128	2304

Table 3 ■

Comparison of Major Cryptographic Algorithms in Image Authentication

Cryptographic Algorithms	Unauthorized Read Detection	Unauthorized Write Detection	Processing Speed	Key Management
Secret Key	+	+	0	−
Public Key	+	+	−	+
Digital Time Stamp	−	+	+	NA

Notations: + means positive (i.e., fast or strong; − means negative (i.e., slow or weak); 0 means average; NA means not applicable.

difficulty of a brute-force attack for private-key lengths.[5] That table states that if one is concerned enough about security to select a private-key algorithm with a 64-bit key, one should choose a module length for the public-key algorithm of about 512 bits.

Algorithm Comparisons

Table 3 summarizes the strengths and weaknesses of the various cryptographic algorithms for improving the security of large image files in digital radiology environments. Because a digital signature can be created by using a public-key method, Table 3 does not evaluate this algorithm.

Digital Image Trust Centers

The purpose of a digital trust center (DTC) is to incorporate systematically a variety of cryptographic algorithms (discussed above) in digital radiology environments. Currently, health care provider organizations lack such a cryptologic model for digital image protection. Figure 4 illustrates the architecture of a digital image trust center within a hospital-integrated PACS environment. In this figure, multimedia data from various image and text sources in a radiology department, such as medical imaging scanners, radiology information systems (RIS), hospital information systems (HIS), individual PAC systems, and film digitizers are linked to a centralized data repository that is managed by the PACS archival server.

In the DTC architecture, the PACS archival server interacts with an authentication server to support the authentication service. The authentication server can attach the hash value and time stamp to an incoming image dataset. The archival server then stores this dataset and its time stamp in the PACS image database and sends a copy of the time stamp back to the imaging source for recording. The image data can be originated by three possible sources. When the imaging source originates from a digital imaging scanner, such as magnetic resonance imaging (MRI) and x-ray computed tomography (CT), the image time stamp should be saved into the database of the acquisition computer of that scanner node. When the imaging source is from a sectional PACS, such as an Ultrasound or Nuclear Medicine PACS, the electronic time stamp should be saved in the local archive of that sectional PACS. When the film digitizer is used to convert external or existing screen films, the time stamp should be saved into the host computer of that film digitizer. Generally, PACS controllers and these acquisition computers can manage time-stamp data through a local database management system (DBMS).

The DTC builds on the infrastructure of the PACS, and so its access control protocols inherited from the PACS. That is, information access is limited to display stations registered in the PACS networks and is granted by patient name or hospital ID only. Also, the PACS does not permit the user to alter the original image and associated data from a display station, although the user can append new findings of an imaging study as separate entries into the central archive. Further work will investigate better access control policies for various kinds of image care services.

Someone who challenges the originality of an image stored in a local imaging site can just query the PACS archival server for verification by providing the image ID. To ease concerns about tampering or backdating at the centralized PACS repository, the scheme can be further refined to blend several sequential time-stamp requests into a chain or a binary tree structure.[12] Such authentication protocols incur little operational overhead with common hash functions; such as Message Digest 4 (MD4) and Message Digest 5 (MD5).[5] Within

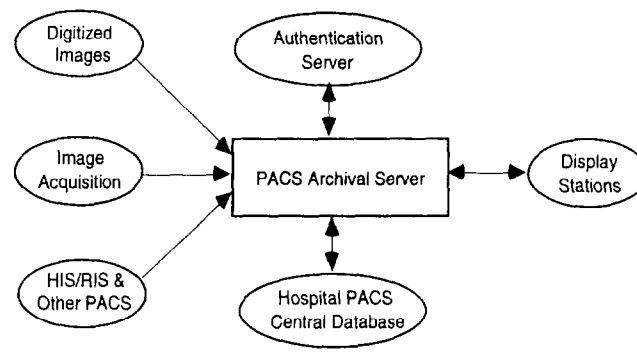

Figure 4 Digital trust center method in PACS for medical image authentication.

the PACS framework, the authentication server can reside in the same computer as the archival center and thus has minimum effect on the overall system architecture. For teleradiology systems without an on-line image database server, an authentication server node should be installed in the expert center site that serves remote small hospitals or rural clinics. The authentication server in this case will calculate and store the electronic time stamps of incoming images.

Moreover, a remote display station can query the PACS server to verify the authenticity of an image received. The display station does so by computing the hash value, or fingerprint, of the image in question and matching this value with the one obtained from the PACS archive. Since massive image transfer is not required, the verification will be done quickly.

The authentication server can also be extended to protect confidentiality. For example, the server creates a random key to encipher an image data file and stores it together with the encrypted image file into the PACS central archive. This assigned key will be sent together with the encrypted image file to the remote display station for confidentiality protection. Such an encryption procedure or protocol, however, can be compromised if the key is intercepted during the transmission. Public-key algorithms are preferable, but they are a thousand times slower than conventional single-key algorithms in encryption. Thus, the encryption protocols in our authentication server used a mix of private and public key cryptographic techniques. Image encryption imposes added difficulties due to the need to store and transmit massive amounts of image data, instead of their hash values or time stamps for authentication. Thus, we devote the next section to discuss the design and performance of encryption protocols in the digital trust center.

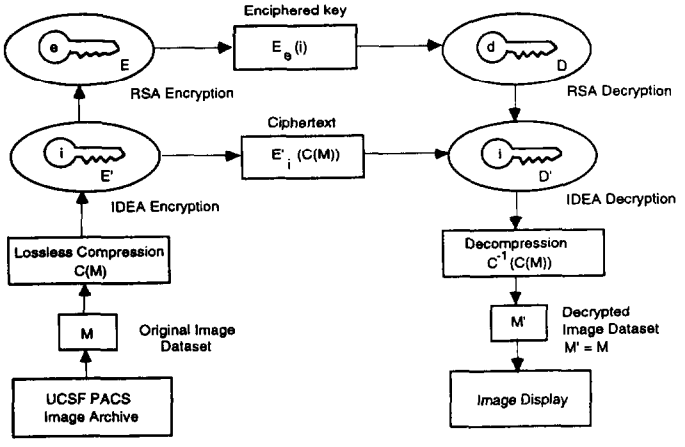

Figure 5 The logical flow of the encryption protocol that incorporates both IDEA secret-key cipher and RSA public-key cipher within a hospital-integrated PACS environment. Medical images are directly retrieved from the PACS central archive. Lossless compression is applied also. Encryption and decryption are performed by two UNIX workstations in the UCSF PACS networks. M'M' = 3D image datasets; C = lossless image compression function; C^{-1} = inverse function of C for lossless decompression; E', E = encryption functions; D, D' = decryption functions; and e, i = encryption keys.

Encryption Protocol

A hybrid encryption protocol has been derived for use in the digital trust center for securing medical image data. This cryptographic protocol takes advantage of the fast encryption of secret-key algorithms and secure key management of public-key methods. This study tested 81 datasets of MRI and positron emission tomography (PET) images retrieved from the UCSF PACS central archive. PET images are converted into 8-bit gray level. The MRI images are divided into 16-bit and 8-bit images. The 8-bit MR images are converted from the heavily T1-weighted scans with no degradation of image quality.

As shown in Figure 5, an image dataset is retrieved from the UCSF PACS archive and subjected to lossless compression. (Since diagnostic and other textual reports are much smaller in size, they are not the focus of this study.) Using data compression together with encryption has three advantages:

- Cryptanalysis relies on exploiting redundancies in the plaintext, and compressing a file before encryption reduces these redundancies.
- Encryption is time-consuming, especially for a large-image file, and compressing a file before encryption speeds up the entire process.
- Transmission time depends on file size, and compressing a file compensates for the expansion due to encryption and reduces the amount of data to be transferred.

It is worth noting that many of the current PAC systems have not yet implemented image compression of any kind. The need for providing image security adds further incentive to incorporate compression into all PAC systems, besides overcoming data storage and transmission limitations. Further, it is important to perform compression before encryption. If the encryption algorithm is any good, the ciphertext will not be compressible; it will look like random data.[5] Thus, most cryptographic packages routinely perform data compression before encryption. The secret-key algorithm known as the International Data Encryption Algorithm (IDEA),[7] is used to encrypt the compressed

images, where the secret key is randomly generated for each image encryption. The popular and robust public-key algorithm known as Rivest, Shamir, and Adleman (RSA),[9] is used to encrypt the randomly generated secret key, i, which is a 1,024-bit number. This public-key–encrypted secret key is sent along with the ciphertext (encrypted image dataset) to the receiver. The receiver uses an individual private key to recover this encrypted secret key and then applies that key to run the fast secret-key algorithm to decrypt the large ciphertext.

In this experiment, the sender and the receiver are located in separate SUN SPARC LX workstations (SUN Microsystems, Mountain View, CA) in our PACS networks. The original and the decrypted image datasets were also evaluated to be equivalent; i.e., there was no contamination to the image datasets during this cryptographic process. The software system uses the Pretty Good Privacy (PGP) package[14] and the UCSF PACS data access and transmission programs.[15,16]

Results

Table 4 summarizes the encryption results on 81 volume image datasets of 16-bit MR (18 samples), 8-bit MR (30 samples), and 8-bit PET (33 samples) images. The 8-bit image datasets are postprocessed with a look-up table implemented in a medical workstation.[17] Two types of performance outcome are measured: the encryption time and decryption time with respect to the entire volume dataset and with respect to individual image slices. All these datasets have a uniform image dimension per slice.

As indicated in Table 4, for a few image slices, or even the entire PET volume dataset, it is feasible to incorporate key-based security software into real-time digital radiology applications (i.e., computational overhead is no more than a few seconds) using such common computing platforms as low-end SUN SPARC workstations.

Table 4 shows transmission with encryption and with decryption over low-end SUN SPARC workstations. Note that the reduction in the size of the ciphertext is due to applying lossless compression on image data before encryption. This leads to improvements in transmission time for encrypted files compared with unencrypted files.

This, however, does not apply to the large-volume data of a brain MRI (average about 7.2 MB) or other, even larger image datasets, such as those of mammograms (10 MB per digitized film) and CT (about 30–40 MB for a chest study). Table 4 shows also that the encryption process is slightly slower than the decryption process in our implementation.

One important use of encryption is to ensure the authenticity and confidentiality of radiologic images in telemedicine applications. Tables 5 and 6 give the transmission time performance of unencrypted and encrypted image datasets and slices with respect to three different communications media: narrow band integrated services digital network (56 Kbs N-ISDN), Ethernet (10 Mbs), and OC3-c asynchronous transfer mode (155 Mbs ATM). Figure 6 shows a set of timing performance charts of the three imaging modalities with and without encryption over these communications networks. To reflect real-life situations, the timing calculation is based on the actual data throughput rates measured in our PACS and teleradiology environment, rather than the maximum bit rates allowed for these communications media.[15,16]

As shown in Figure 6, the faster transmission speed of encrypted files is largely due to the lossless compression done before the encryption process. Encryption normally results in similar file size. For low-speed communications media such as N-ISDN, and in the absence of data compression, the hybrid encryption

Table 4 ■

Time Performance of Encryption and Decryption Based on the Hybrid Scheme in Figure 5

Modality	Image Dimension	Average Volume Size (MB)	s.d.	Average Ciphertext Size (MB)	s.d.	Average Encryp./Decryp. Time per Volume (s)	s.d.	Average Encryp./Decryp. Time per Slice (s)	s.d.
MRI—Brain	256 × 256 × 16-bit	7.21	±0.51	2.51	±0.68	239.17	±26.46	4.24	±0.32
						184.61	±21.47	3.28	±0.27
MRI—Brain	256 × 256 × 8-bit	3.57	±1.20	1.73	±0.72	100.91	±16.90	1.92	±0.23
						88.06	±10.42	1.68	±0.15
PET—Brain	128 × 128 × 8-bit	0.75	±0.00	0.36	±0.15	23.43	±4.88	0.50	±0.10
						20.03	±2.37	0.43	±0.05

Notations: s.d. = standard deviation; MB = megabytes; s = seconds.

Table 5 ■

Effective Transmission Performance of Encrypted and Unencrypted Image Datasets Over N-ISDN (23 Kbs), 10-Base-T Ethernet (1 Mbs), and OC3-c ATM (65 Mbs) Networks

Imaging Modality	Mean Volume Size (MB)	Mean Ciphertext Size (MB)	Communication Medium (averaged data throughput)	Mean Transmission Time per Unencrypted Volume (s)	s.d.	Mean Transmission Time per Encrypted Volume (s)	s.d.
16-bit MRI	7.21	2.51	N-ISDN	2508.06	±176.11	872.55	±237.52
			Ethernet	57.69	±4.05	20.07	±5.46
			ATM	0.89	±0.06	0.31	±0.08
8-bit MRI	3.57	1.73	N-ISDN	1242.56	±417.67	601.16	±251.62
			Ethernet	28.58	±9.61	13.83	±5.79
			ATM	0.44	±0.15	0.21	±0.09
8-bit PET	0.75	0.36	N-ISDN	261.565	±0.00	124.182	±53.884
			Ethernet	6.016	±0.00	2.856	±1.239
			ATM	0.093	±0.00	0.044	±0.019

Kbs = kilobits per second; Mbs = megabits per second.

scheme described in Figure 5 not only ensures data integrity but also provides faster transmission. This is a useful finding, as most teleradiology applications are using low-speed public networks or the Internet for transmission. For high-speed broadband networks of 1 Mbs or more, however, there is a noticeable timing performance degradation when using encryption software, either with or without compression. For instance, Table 5 shows that the mean transmission time per unencrypted (original) 16-bit MRI volumetric dataset is 57.69 seconds (s), with a standard deviation of 4 s. In comparison, the mean transmission time and mean encryption time per encrypted 16-bit MRI dataset is: 20.07 s + 239.17 s = 259.24 s (from Tables 4 and 5), with a standard deviation of 5.46 s + 26.46 s = 31.92 s. The PACS networks are usually built on broadband technology for fast image transmission, so software implementation of image encryption thus incurs noticeable timing overhead. Therefore, the decision whether to apply encryption in a digital radiology environment becomes a tradeoff between time and security.

Discussion

Ready access to medical documents in the coming era of digital radiology carries with it the responsibility for ensuring that information is both authentic and confidential. The growing use of digital medical images in clinical practice poses new security issues due to the large image size and different user requirements. Research in computer cryptography has reached such a level of maturity that many robust algorithms are now available. Recently, papers discussing the use of specific cryptographic techniques in authenticating medical images have also been published. What is still missing, however, is the understanding

Table 6 ■

Effective Transmission Performance of a Single Encrypted and Unencrypted Image Slice Over N-ISDN (23 Kbs), 10-Base-T Ethernet (1 Mbs), and OC3-c ATM (65 Mbs) Networks

Imaging Modality	Mean Volume Size (MB)	s.d.	Mean Ciphertext Size (MB)	s.d.	Communication Medium (estimated throughput)	Transmission Time per Unencrypted Slice (s)	Mean Transmission Time per Encrypted Slice (s)	s.d.
16-bit MRI	7.21	±0.51	2.51	±0.68	N-ISDN	45.59	15.49	±4.22
					Ethernet	1.05	0.36	±0.10
					ATM	0.02	0.01	±0.001
8-bit MRI	3.57	±1.20	1.73	±0.72	N-ISDN	22.80	11.47	±4.80
					Ethernet	0.52	0.26	±0.11
					ATM	0.008	0.004	±0.002
8-bit PET	0.75	±0.00	0.36	0.15	N-ISDN	5.699	2.642	±1.15
					Ethernet	0.131	0.061	±0.03
					ATM	0.002	0.001	±0.0004

Note: Kbs = kilobits per second; Mbs = megabits per second.

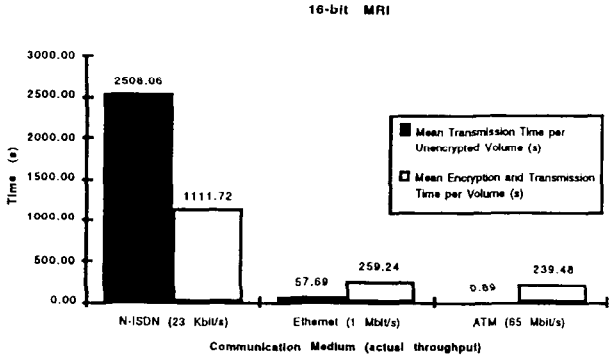

Figure 6.a

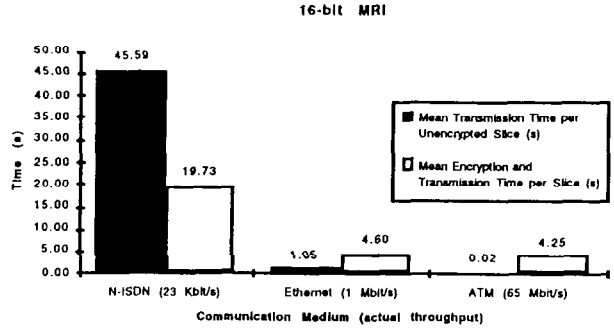

Figure 6.b

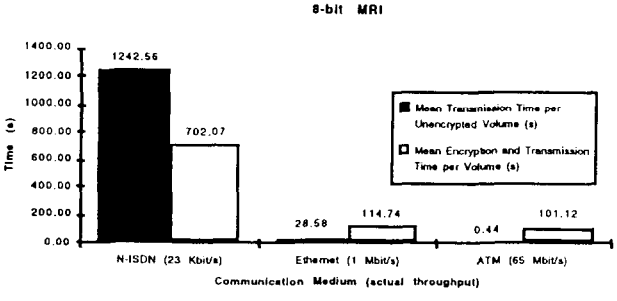

Figure 6.c

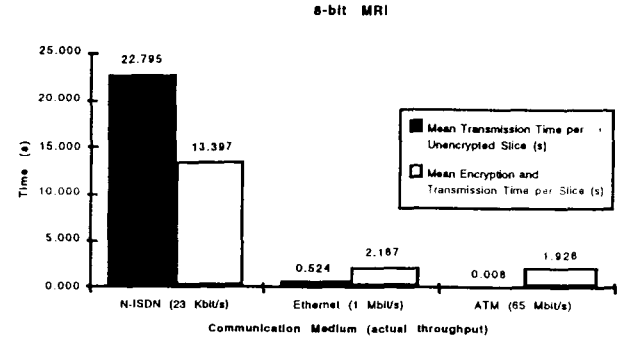

Figure 6.d

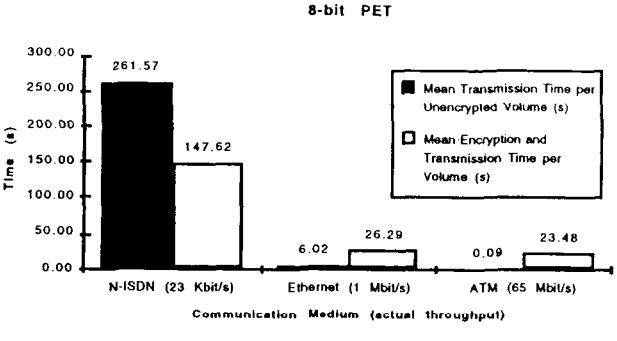

Figure 6.e

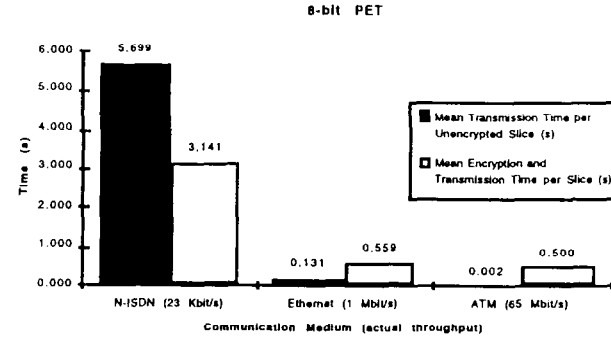

Figure 6.f

Figure 6(A–F) Transmission time performance charts of three types of image files and slices with and without encryption.

of how to integrate these cryptographic algorithms coherently into the existing operations of the enterprise.

This study takes a software system view of creating trust in digital radiology images and investigates the holistic approach of integrating a spectrum of cryptographic algorithms to meet the various security requirements of patient records. The key idea is to introduce a digital trust center as a digital notary to certify and protect the confidentiality of large volumes of medical images in hospitals. We evaluated the performance of our cryptosystem in encrypting several imaging modalities: 16-bit MRI, 8-bit postprocessed MRI, and PET, using cryptographic protocols operated within the hospital's integrated PACS environment. We presented the experimental results based on the sample size of 81 volumetric MR and PET images archived in our PACS repository. Future work will experiment with other encryption techniques to show performance comparisons.

Our result indicates that image authentication can be readily incorporated into the existing PACS architecture without affecting existing operations. Real-time software encryption to protect confidentiality is possible for one or a few selected image slices but not for the entire volumetric dataset (except the low-volume datasets, such as PET images). The encryption of 16-bit MRI and other higher imaging modalities, such as CT and mammography, would require dedicated hardware. Besides the considerable costs, the lack of interface standards also makes the current generation of cryptographic hardware not readily portable across different digital radiology systems.

Over low-speed phone lines, ISDN, and the Internet, the hybridization of lossless compression, public-key cryptography, and secret-key cryptography can reduce the transmission time greatly while providing added security and quality assurance. Certainly, lossy compression can be interchanged with the lossless scheme to further speed up the transmission, but this is done at the expense of image quality.

Many protocols for non-medical applications have been developed recently for securing and authenticating digital information. For example, the Internet Engineering Society developed the Privacy Enhanced Mail (PEM) standard software, which utilizes DES to encrypt e-mail text and RSA to authenticate and sign it. ViaCrypt (Lemcom, Phoenix, AZ) and PGP use IDEA instead of DES for encryption. Digital Notary (Surety Technologies, Morristown, NJ) uses tree-based digital time stamping for data authentication.[11] These protocols can be easily incorporated into the digital trust center model and optimized for digital radiology applications. In addition, most encryption software packages come with automatic key management software that contains files of public- or secret-key material, the owner's user ID, and a time stamp showing when a certain key pair was generated. Often, the secret key files are encrypted with their own passwords or pass phases for protection.

Although cryptography is a powerful system for protecting medical data, it is not a panacea. For instance, cryptography can not protect against stolen encryption keys. The whole point of using encryption is to make it possible for people who have your encryption keys to decrypt your files. Thus, any attacker who can steal your keys can decrypt your files. Also, cryptography cannot protect against destructive attacks. Sometimes, an attacker does not want to read your files but just wants to keep you from reading them. Even an attacker who cannot get access to your encryption keys can still cause you a lot of pain and suffering by breaking into the image archive and erasing relevant medical documents. To solve such problems, the health care provider organizations must enforce proper protocols or procedures for access control.[18] A rudimentary form of access control, inheriting the current data access protocol of PACS, is provided in the digital trust center model. Future work will investigate this important security issue for more specialized image care services in more detail.

In the past, the whole issue of cryptography was clouded by disputes over export and government access for law-enforcement purposes. The discovery and publication of public-key algorithms and digital time stamping open up many vistas for using robust cryptography in non-federal sectors. These two kinds of cryptographic algorithms are the core techniques to establish the trust in multimedia medical records among hospitals using digital images. Research should now focus on the coherent integration of these algorithms into existing hospital information systems.

Nevertheless, there can never be a purely technologic soluton to privacy, and social issues must be considered in their own right. The digital trust center provides a systematic approach for integrating various cryptographic techniques and constructing computer systems that are privacy enabled, giving power to the individual or local hospital. But only the medical community's proper appreciation for these techniques and an understanding of their interrelations can cause the right cryptosystem to be used.

The author thanks Mr. Macro Abundo of the San Francisco VA Medical Center for his assistance in software implementation and data analysis. He also thanks Dr. H. K. Huang and Dr. Ronald Arenson of the Department of Radiology, UCSF, for informing him of the security issues in digital radiology departments. Their practical concerns motivated this investigation and development.

References ■

1. Huang HK, Ratib O, Bakker AR, Witte G. NATO ASI Series F, PACS in Medicine, vol. 74. New York: Springer–Verlag, 1991.
2. Osteaux M (ed). A Second Generation PACS Concept. Berlin: Springer–Verlag, 1992.
3. Huang HK, Arenson RL, et al. Second generation PACS at UCSF. Radiology. 189:1993;290.
4. Smith JP. Authenticating digital medical images with digital signature technology. Radiology. 194:1995;771–4.
5. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. New York: John Wiley & Sons, 1993; Chapter 7.
6. Branstad DK, Gait J, Katzke S. Report on the workshop on cryptography in support of computer security. NBSIR 77-1291, National Bureau of Standards, Sept. 21–22, 1976.
7. Lai XJ. On the design and security of block ciphers. In: Massey JL (ed). ETH Series on Information Processing, vol. 1. Konstanz, Switzerland: Hartung-Gorre Verlag, 1992.
8. Diffie W, Hellman ME. New directions in cryptography. IEEE Trans Information Theory. 22:1976;644–54.
9. Rivest R, Shamir A, Adleman L. A method for obtaining digital signatures and public-key crytosystems. Communications of the ACM. 21:1978;120–6.
10. Davies DW, Price WL. Digital signature—An update. Proceedings of the International Conference on Computer Communications, Sydney, Oct. 1984. North Holland: Elsevier, 1985;843–7.
11. Cipra B. Electronic time-stamping: The notary public goes digital. Science. 261:1993;162–3.
12. Haber S, Stornetta WS. How to time-stamp a digital document. Journal of Cryptography. 3:1991;99–112.
13. Wiener MJ. Efficient DES key search. TR-244, School of Computer Science, Carleton University, May 1994.
14. Garfinkel S. PGP—Pretty Good Privacy. O'Reilly & Associates, CA, 1995.
15. Wong STC, Huang HK. A hospital integrated framework for multimodal image base management. IEEE Trans. Systems, Man, and Cybernetics. 26:1996;455–69.
16. Wong STC, Huang HK. Limitations of transmission control protocol on high-speed radiologic asynchronous transfer mode networks. Radiology. 197(P):1995;258.
17. Wong STC, Knowlton RC, Hawkins RA, Laxer KD. Multimodal image fusion for noninvasive epilepsy surgery planning. IEEE Computer Graphics & Applications. 16:1996;30–8.
18. Varadharajan V, Calvelli C. An access control model and its use in representing mental health application access policy. IEEE Trans. Knowledge & Data Engineering. 8:1996,81–95.

UNITED STATES POSTAL SERVICE

Statement of Ownership, Management, and Circulation
(Required by 39 U.S.C. 3685)

1. Publication Title	2. Publication No.	3. Filing Date
Journal of the American Medical Informatics Association	0 0 1 1 - 5 6 5 0	9/30/96
4. Issue Frequency	5. No. of Issues Published Annually	6. Annual Subscription Price
bimonthly	6	$102.00

7. Complete Mailing Address of Known Office of Publication *(Street, City, County, State, and ZIP+4) (Not Printer)*
American Medical Informatics Association, 4915 St. Elmo Ave., Suite 302, Bethesda, MD 20814

8. Complete Mailing Address of Headquarters or General Business Office of Publisher *(Not Printer)*
210 S. 13th Street, Philadelphia, PA 19107

9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor *(Do Not Leave Blank)*
Publisher *(Name and Complete Mailing Address)* Hanley & Belfus, Inc., 210 S. 13th St., Philadelphia, PA 19107 for the American Medical Informatics Assn., 4915 St. Elmo Ave., Suite 302, Bethesda, MD 20814

Editor *(Name and Complete Mailing Address)*
William W. Stead, MD
2209 Garland Avenue South
Nashville, TN

Managing Editor *(Name and Complete Mailing Address)*
Sandra Lovegrove, Hanley & Belfus, Inc., 210 South 13th Street Philadelphia, PA 19107

10. Owner *(If owned by a corporation, its name and address must be stated and also immediately thereafter the names and addresses of stockholders owning or holding 1 percent or more of the total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.) (Do Not Leave Blank.)*

Full Name	Complete Mailing Address
American Medical Informatics Association	4915 St. Elmo St., Suite 302 Bethesda, MD 20814

11. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages or Other Securities. If none, check here ☒ None

Full Name	Complete Mailing Address

12. For completion by nonprofit organizations authorized to mail at special rates. The purpose, function, and nonprofit status of this organization and the exempt status for federal income tax purposes: *(Check one)*
☐ Has Not Changed During Preceding 12 Months
☐ Has Changed During Preceding 12 Months
(If changed, publisher must submit explanation of change with this statement)

13. Publication Name: Journal of the American Medical Informatics Association	14. Issue Date for Circulation Data Below: July/August 1996	
15. Extent and Nature of Circulation	Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies of Single Issue Published Nearest to Filing Date
a. Total No. Copies *(Net Press Run)*	3,932	4,000
b. Paid and/or Requested Circulation (1) Sales Through Dealers and Carriers, Street Vendors, and Counter Sales *(Not Mailed)*	0	0
(2) Paid or Requested Mail Subscriptions *(Include Advertisers' Proof Copies/Exchange Copies)*	3,547	3,508
c. Total Paid and/or Requested Circulation *(Sum of 15b(1) and 15b(2))*	3,547	3,508
d. Free Distribution by Mail *(Samples, Complimentary, and Other Free)*	54	54
e. Free Distribution Outside the Mail *(Carriers or Other Means)*	0	0
f. Total Free Distribution *(Sum of 15d and 15e)*	54	54
g. Total Distribution *(Sum of 15c and 15f)*	3,601	3,562
h. Copies Not Distributed (1) Office Use, Leftovers, Spoiled	331	438
(2) Return from News Agents	0	0
i. Total *(Sum of 15g, 15h(1), and 15h(2))*	3,932	4,000
Percent Paid and/or Requested Circulation *(15c / 15g x 100)*	98.5%	98.5%

16. This Statement of Ownership will be printed in the Nov/Dec 96 issue of this publication. ☐ Check box if not required to publish

17. Signature and Title of Editor, Publisher, Business Manager, or Owner: Sandra G. Lovegrove, Production editor — Date: 9/20/96

I certify that all information furnished on this form is true and complete. I understand that anyone who furnishes false or misleading information on this form or who omits material or information requested on the form may be subject to criminal sanctions *(including fines and imprisonment)* and/or civil sanctions *(including multiple damages and civil penalties)*.