Security for the Digital Information Age of Medicine: Issues, Applications, and Implementation

Michael A. Epstein, Michael S. Pasieka, William P. Lord, Stephen T.C. Wong, and Nicholas J. Mankovich

Privacy and integrity of medical records is expected by patients. This privacy and integrity is often mandated by regulations. Traditionally, the security of medical records has been based on physical lock and key. As the storage of patient record information shifts from paper to digital, new security concerns arise. Digital cryptographic methods provide solutions to many of these new concerns. In this article we give an overview of new security concerns, new legislation mandating secure medical records and solutions providing security.

KEY WORDS: security, medical records, cryptography, teleradiology, digital signatures, certificates, RSA, smartcard, radiology, computers

1. INTRODUCTION

TECHNOLOGY can meet increasing demands for secure digital medical information arising from patients, policy makers, and others.[1,2,3] Banking and the military already use security technologies. This article is an overview and survey of security as applied to medicine and covers the changing needs for security, available technologies, and implementation considerations.[4]

In medical environments before the computer, the security of records relied on physical lock and key. Many computer-based systems have individual users interacting with specific information gathering or presentation applications. Security for these systems can also rely on physical lock and key, but usually implements locks through passwords. With today's networking technology as an enabler, managed healthcare strives for efficiency and broader access to medical information. In this networked environment a whole new paradigm for security must be implemented, as multi-facility health maintenance organizations (HMOs) and information hungry community health information networks (CHINs) are increasingly relying on insecure networks, like the internet, to exchange what should be private, immutable and verifiable medical information.

From the perspective of a healthcare consumer, privacy of medical information can be very important. Insurance or job opportunities may be denied if news of a patient's medical condition is discovered. Public figures may get unwanted publicity should medical information be released. Patients with medical conditions having an associated social stigma will not want knowledge of their condition disseminated or they may wish anonymous access to disease related information. Federal and state laws are beginning to treat many aspects of security in medical practices.

We begin this article by introducing the major issues that are involved in providing a secure medical environment, with reference to the techniques used to provide such security. Next, we walk through the protocols used to insure security. An overview of the mathematical basis of various security techniques follows.[5,6,7] We then assess the true security of the various methods and estimate the overhead they impart on system and user performance. Finally, some of the legal requirements of healthcare providers to provide security, legal restrictions on the use of security technology, and a practical example demonstrating the need for careful implementation of security functions are presented.

2. INTRODUCTION OF TERMS

Security Objectives

Security is a complex objective. To successfully achieve this objective the issues and terms of security must first be defined.

Authentication: Are users who they say they are?

Access Privileges: What applications and information can a particular user read, write, or modify?

Privacy: Can the information be understood by someone other that the intended recipient?

Immutability: Can the information be modified without detection?

Accountability: Did the person responsible for the information *really* sign it?

From Philips Research, Briarcliff Manor, NY, and Palo Alto, CA.

Address reprint requests to Michael A. Epstein, Philips Research, 345 Scarborough Rd, Briarcliff Manor, NY 10510-2099.

0897-1889/98/1101-0004$8.00/0

Traceability: What is the change and review history of the information?

Origination: Where, or on what system, was the information created or captured? (This is not a major issue in the medical domain and will not be covered further. In areas where this is a concern there exists digital watermarking techniques to help provide solutions.)

Cryptographic Terminology

Plaintext or Cleartext: A text message or other data (including images) before encryption. This is any kind of digital message noted as m.

Ciphertext: The plaintext after encryption noted as c.

K: A secret key used to encrypt or decrypt data. For digital data, K is a number. Where needed, an optional subscript distinguishing the type of key, private or public, and an owner, is added. For example, $K_{private\text{-}bob}$ signifies the private key for Bob.

Encryption: A mathematical function that converts a plaintext message to an unreadable form using K noted as $c = E(K,m)$.

Decryption: A function that performs the inverse of the encryption function, noted as $m = D(K,c)$.

Hash: A mathematical function that will generate a fixed length representation, or fingerprint, of a document.

Communications Model

A communications model cryptographers use to analyze security concerns is shown in Fig 1. Each of the participants in an exchange have traditional names to refer to them:

- *Client:* An approved party wishing to obtain, modify or add to information contained on a server. In the medical domain this would be a medical professional, ie, a physician, nurse, or administrator.
- *Server:* A repository of needed information. In the medical domain this could be the hospital's central computer system.
- *Trent:* A trusted third party. A person or computer system that can be trusted by both the client and the server without reservation. In a secure system this party serves as the basis for trust over a network.
- *Eve:* An attacker that can eavesdrop on communication between the client and server. It is assumed that Eve can read all bits transmitted over the network. Eve also has the ability to store and analyze these bits.
- *Mallory:* An attacker that is more powerful that Eve. Like Eve, Mallory can see all the bits on the network, but can also delete, modify, or add bits in the transmission of any bits on the network.

The assumption of abstract powerful attackers aids in building secure systems. It is clear that if attackers such as Eve and Mallory can be defeated by the security of a system, then more realistic attacks will also fail.

3. CONCEPTS FOR SECURE SYSTEMS

When a client contacts a server and requests a secure connection, there are two separate issues to address: protocol and encryption. Protocol describes the steps necessary to establish a secure connection between two parties and the steps used to pass messages securely. Encryption is the mathematical means used to disguise a message so as to hide its contents from unauthorized viewing. A discussion of the mathematics of one popular

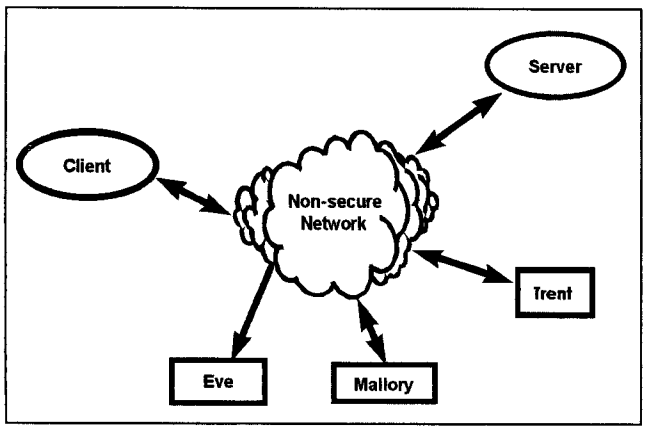

Fig 1. A communications model showing a client and a server attempting secure communications over an insecure network with Trent, a helper, and Eve and Mallory as attackers.

method of encryption, RSA, (for Rivest, Shamir, Adelman) is discussed in section 4.

A significant early step in any protocol used for the establishment of a secure connection is the agreement of the server and client on the type of encryption to use. Modern cryptographic systems rely only on the security of the key. It is assumed that an attacker will obtain knowledge of the cryptographic algorithm and the protocol. (For further discussion of attacks on supposedly secure systems, see section 7.)

There are two major types of encryption currently in use: symmetric key and public key. Each has strengths and weaknesses. In practice, most protocols use a combination of both types employing the strengths of both. (For an example of this, see section 6.)

The basic protocol used by both systems is straightforward. The message, also termed plaintext, is encrypted through a mathematical method into ciphertext. The ciphertext is then transmitted over a channel that is assumed to be insecure. The ciphertext is then decrypted through an inverse mathematical method back into the original plaintext message.

Symmetric Key Encryption

In a symmetric key encryption system there is only one key used for both the encryption and decryption of a message. The strength of this method is the relatively fast speed of the encryption and decryption algorithms when implemented in computer software. Further speed increases can be achieved by the use of special purpose hardware. This makes the use of symmetric key encryption attractive for bulk transfers of information.

The major drawback of a symmetric key scheme (like the one shown in Fig 2) is the problem of key distribution. If two parties desire to communicate securely, a single key must be generated in a secure manner in one location and then be securely transported to a second location for use. This example of distribution makes two assumptions: there are only two parties who must securely communicate and the key will never be compromised by either of the two parties.

In situations where many parties each need independent secure communication with each other, key distribution is a major problem. For example, if there are ten parties, each must have a separate key to communicate with the other nine. The total number of keys, which must be generated and correctly distributed, is n(n-1)/2 or, in this case, 45. Adding one more party to the group means generating 10 more keys and securely distributing one to each of the other ten parties. In any large system, this problem rapidly grows out of control. Not only does key distribution become a problem, but so does remembering the keys of all people that one must communicate with. Unlike telephone numbers, keys should not be posted, as security would be compromised. A secure database of keys would have to be developed for each party.

Public Key Encryption

In a public key encryption system, there are two mathematically related keys. Encryption and decryption are asymmetric. That is, one cannot use one key to both encrypt a message and decrypt the resulting ciphertext back to plaintext. As shown in Fig 3, given a plaintext message, one key is used to encrypt to ciphertext and the other key is used to recover the plaintext. At the time of key generation, there is no difference between the two keys. One is selected to be the public key and the other to be the private key. The public key is published so that it can be known to anyone. The private key is kept secret in a secure manner. It must be very difficult to discover the private key from the public key. One manner of keeping a private key secure is by the use of smartcards which are discussed in section 5.

The weakness of the symmetric key system, key

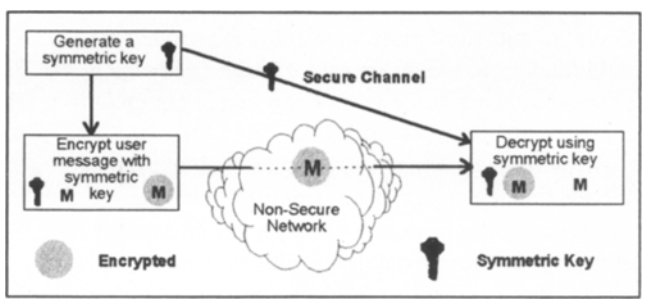

Fig 2. Symmetric key encryption using a secure channel to distribute a key, and using a non-secure network to distribute messages encrypted with the key.

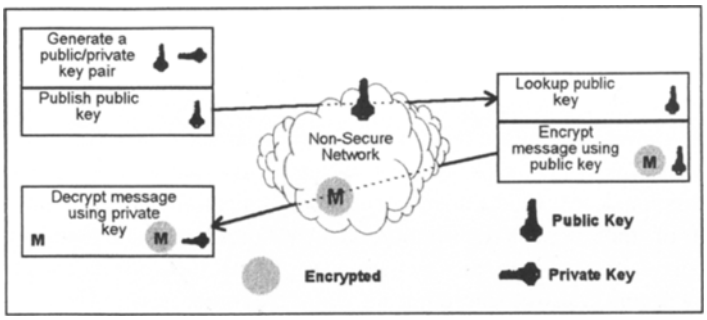

Fig 3. Public key encryption allows a non-secret key to be passed over any network. That key can then be used to encrypt and send a message that only the holder of the private key can decrypt.

distribution, is not a problem in the public key system. Here, one wants to publish the public key in such a way that those parties wishing to securely communicate will be guaranteed to have the correct public key and not the public key of some other party. One way of accomplishing this is by use of certificates, as described later in this section. Additionally, the number of keys (or in this case public/private key pairs) that must be generated is one per server and not one per independent connection. The major weakness of the public key system is the relative slowness of the method when implemented in computer software or hardware. In comparison to symmetric key systems, public key systems are between ten and one hundred times slower.

Digital Signatures

One use of public key encryption is in creating digital signatures. Signatures found on physical documents have certain properties that must be carried over to digital signatures. The two primary properties that physical signatures have are immutability and accountability. These two properties can be further elaborated:

- Authenticity
- Unforgability
- Nonreusability
- That the signed document cannot be altered without detection
- The signature can not be repudiated

A digital signature must have all of the properties. Additionally, it would be useful if the verification of the digital signature is easy and fast to compute, however this is not as crucial.

To assist in ensuring the properties of nonreusability and that the signed document cannot be altered, the concept of a hash of a document is introduced. A straightforward analogy is that a hash of an electronic document is similar to a human fingerprint. A hash uses a mathematical function that has the following three properties:

1. Impossible to take the hash value and recover the document
2. Nearly impossible to find a second document that hashes to the same value
3. Changing one bit or character of the document will change on average fifty percent of the bits of the hash value

The first property holds simply because a hash involves a loss of information. Normally, the hash value is a fixed length of, for example, 128 bits. Most documents are significantly longer than 128 bits (equivalent to 16 characters). The second and third properties are dependent on the mathematics of each particular hash algorithm. Additionally, the function should be relatively simple and fast to compute in software.

Referring to Fig 4A, the process of creating a digital signature is straightforward:

- Compute the hash value of the document
- Encrypt the hash value with the private key of the person signing the document

If both the hash and public key encryption methods used are secure then all of the objectives of a digital signature mentioned above will hold.

Referring to Fig 4B, the process of verifying a digital signature is also straightforward, given that individuals have what they believe to be an exact copy of the original electronic document, knowledge of the hash method used, the digital signature, and the public key of the person who digitally signed the document. Verifying requires the following steps:

- Compute the hash of the document
- Decrypt the signature
- Compare the computed hash with the decrypted signature

If there is an exact match, then it is proven that both

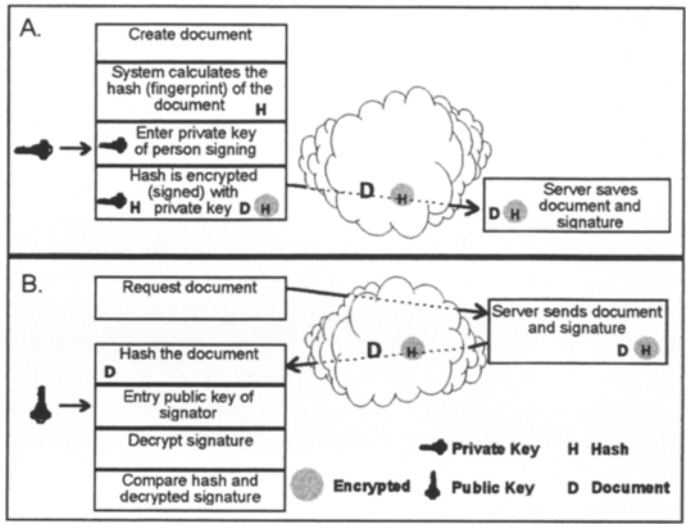

Fig 4. A digital signature is created by hashing a document and then encrypting it. Later the signature and document can be verified.

the signature is verified and the copy of the document is identical to the original.

Certificates

When paper documents are signed for legal purposes a trusted party countersigns the document to guarantee that the signatures are valid. A certificate is a plaintext document containing information that is digitally signed by a trusted third party. Normally, the information contained in a certificate must be verified unaltered before use. To make it possible for the client to verify the certificate, a standard set of information is included in the certificate. This set includes the name of the company that signed the certificate, method of hashing, method of encryption and dates of validity of the certificate.

In the authentication protocol, there is a slight chicken and egg problem: the client must have the public key of the trusted third party that signed the certificate. In the case of World Wide Web (WWW) browsers, this is done by building into the software the public keys of major companies selling certificates. An additional piece of information that is useful to put in a certificate is the public key of a server. The client connecting to the server wants to verify that the server responding is the correct one and not some other party trying to mimic the server and gain secret information from the client.

In Fig 5, the authentication is performed as follows:

- The client requests the certificate of the server from the server of interest.
- The software looks up the public key of the certifying company named in the certificate
- The client verifies the signature of the certificate

The software can then use the public key of the

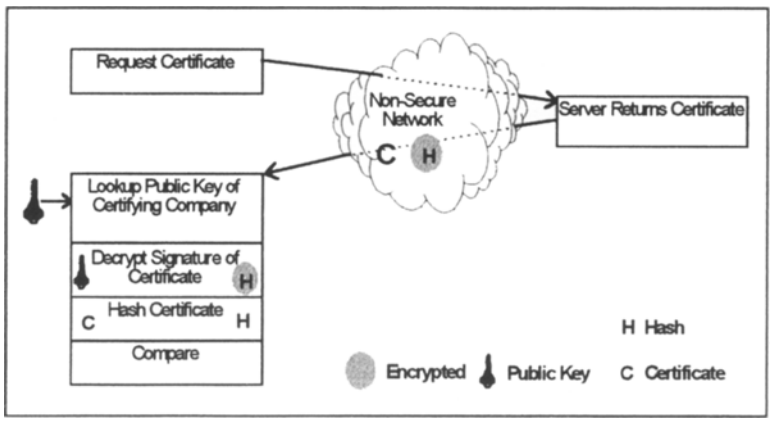

Fig 5. Authentication of a certificate using a digital signature.

server contained in the verified certificate to establish a secure connection. The protocol for establishing a secure connection is described in section 6.

Timestamps

A timestamp is a certificate containing the time when a specific event occurred. This can be used in combination with a digital signature to specify when a document was signed. Specifically, once a document is digitally signed, the signature is timestamped. To verify a timestamp, one must go through the same process as verification of a certificate since a timestamp is a particular kind of certificate. A record of all timestamps issued by a particular trusted third party is kept to prove, if necessary, that the issuers integrity has not been compromised.

The timestamp can then be used to verify not only that a document was signed, but that it was signed at a particular time. This extends the properties of a digital signature, which states that a signature can not be repudiated. If the document was not timestamped, it would be possible for a party to create a document and digitally sign it, only later to revise the document and resign. Having the timestamp as an additional check prevents this potential type of fraud.

4. MATHEMATICS OF A PUBLIC KEY SYSTEM

As previously discussed, one of the methods used in many cryptographic systems is public key cryptography. In a public key system two keys are created that have a mathematical relationship. One of the keys (it does not matter which one) will be designated the private key $K_{private}$. This key is never shown to anyone and is never transmitted over a network. The other key is designated the public key, K_{public}, which can be shown to anyone. If two parties, Alice and Bob, wish to communicate, they each create a public/private key pair and send their respective public key over a network to the other party, or each puts their key in a public directory. Alice can then encrypt any message using Bob's public key: $E(K_{public\text{-}bob}, m)$. Bob can then decrypt the message using his private key $m = D(K_{private\text{-}bob}, E(K_{public\text{-}bob}, m))$. Only Bob can read the message since only Bob has the private key. Even Alice, who created the message, cannot read it once she has encrypted it. If Bob wishes to send a message back he will use Alice's public key.

For a public key system to work it must be very difficult to read an encrypted message without the private key, even if many messages are gathered and substantial computer power is applied using brute force methods to discover the private key (see the section 7). It must also be difficult to discern the private key given the public key. Functions that prevent finding the private key from a public key or decrypting a message are called trapdoor one-way functions.[8] They are one-way because it is relatively easy to compute F(x) given any x, but relatively difficult to compute x given F(x). An encryption function E(m) where m is a message is a one-way function. The trapdoor part of the function states that given F(x) and some secret s, it is easy to compute $x = F^{1}(s, F(x))$. Thus the inverse function for the decoding of a message is $m = D(K, E(m))$, where the key K is known. To form the basis of a trapdoor one-way function a historically difficult problem is used. It is assumed that because mathematicians have tried to solve this problem over a long period of time that the problem is difficult. There is no proof that any trapdoor one-way functions exist.

The public key system described here, called RSA[9], is named after the inventors Rivest, Shamir and Adelman. The difficult mathematical problem they used is called the factoring problem: Choose two prime numbers p and q. Multiply p and q to produce a result n. It is easy to multiply the two numbers, but difficult to factor the result. This problem is well over 2000 years old. To form a cryptographic system the following steps are performed:

1. Choose a random number e such that the GCD(e, (p-1)(q-1)) = 1. This means that e is relatively prime to (p-1)(q-1). The GCD function finds the greatest common denominator of two integers.
2. d is computed such that ((e)(d)) mod ((p-1)(q-1)) = 1. The mod function finds the remainder of a division between two integers.
3. The numbers n and e are the public key. The number d is the private key. The numbers p, q, (p-1), and (q-1) are destroyed.
4. To encrypt a message m we compute the ciphertext c such that $c = m^e \bmod n$.
5. To decrypt the ciphertext we compute: $m = c^d \bmod n$.

The foregoing methods described the mechanics of the RSA public key cryptosystem. The next consideration is the security of the system. If p and q are

large, where large means greater that 1024 bits (about 260 decimal digits), then the factoring of n is an intractable problem even if great computer power is used to search for the answer. However, this remains the best known approach to finding m or d. The numbers p and q must be prime, but methods exist that will find prime numbers in a reasonable period of time. The numbers p, q, e and d need only to be found once. There are also a very large number of primes available for use.[10] The RSA system is the most widely used public key system and has been in use for more that 15 years.

5. THE SECURITY OF SMARTCARDS

When the authentication of an individual by a secure system becomes an issue, one of the most effective means of assuring security is by the use of a smartcard. A smartcard can provide a secure method of binding a given user to a public/private key pair. In this context the smartcard serves as a high tech key chain. Because the user must know a large number that represents the private key, the safest place to keep this key would be in the user's head. Because most of us do not have the ability to memorize 260 digit numbers, a storage device of some type must be used. One solution is to store the number on a computer system. This is not secure since many users share computers and could conceivably find another user's key. Certainly system administrators can read every file on the system. Therefore, the safest place to store a key is on the user's person much like a conventional physical key. This leaves two areas of concern. It is possible that when a key is entered into the computer through a card reader a malicious program could steal the key. To avoid this, smartcards contain not only a private key, but also have a computer that can be used to perform all necessary computations that involve the private key. Therefore, the private key never leaves the card and remains secure.

The second concern is the possibility that someone could steal the physical card. A smartcard, like a bank card, requires activation with a personal identification number (PIN) code that the user can memorize. If the card is stolen, only three incorrect attempts to activate the smartcard are allowed. After these attempts the smartcard will shut down. The smartcard is also protected against a variety of physical intrusions including reverse engineering of the card. It is of course possible to steal both the PIN code and the physical card, but this is a difficult task at best. In the future smartcards may be coupled with fingerprint or retina sensors so as to confirm the presence of the correct user when the card is used.

A smartcard so constructed solves a number of basic difficulties.

1. The long numbers representing keys will not be written down anywhere.
2. Records can only be accessed by authorized users that have both a PIN code and a physical card.
3. The terminal or computer used to access secure information need not be secure because the cryptographic operations involving the key will be performed in the smartcard.
4. Users cannot repudiate their signature by a priori, anonymously publishing their private key; they cannot access the key any better than an attacker who has stolen the card.

6. SOME REAL WORLD EXAMPLES

Consider the following scenario for a design of a medical system for patient information. The system must allow referring, performing, and reviewing physicians access to a database of patient records. Part of this access should allow for a radiologist to file a digitally signed report on the condition of a patient, based on the display of images accessed from the database. The images should be digitally signed, as well, to prevent modification. A WWW server for the display of information by a WWW browser client is to be used to guarantee portability of the system across a wide variety of machine types. The WWW server should dynamically create and serve pages based on query requests of the client. A connection to a database is used to access the report and signature as well as images.

Given the scenario, we need the use of a secure connection between the client and server as well as a method for digitally signing the report filed by the radiologist. For the secure connection we can use a well known software package called Secure Sockets Layer (SSL, Netscape, Sunnyvale, CA).[11] This set of code is incorporated into all major WWW servers and client browsers. The protocol used results in a one way authentication, the client authenticates the server, as well as a secure connection. SSL uses the speed of symmetric key encryption for bulk transfer of data once a secure connection is established, and the security and ease of key distribution via public key encryption for passing

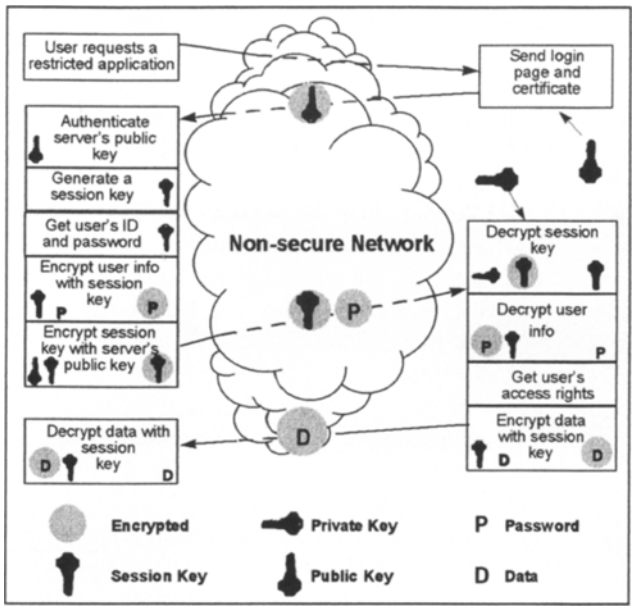

Fig 6. Protocol for establishing a secure connection between a client and a server by using a public key method to exchange a symmetric key.

the symmetric key. Referring to Fig 6, the protocol for establishing a secure connection is as follows:

- The client authenticates the server of interest by requesting a certificate from the server and validating it.
- The client generates a random symmetric key to be used for the bulk transfer of data during the connection. This random key is called the session key. The generation of this key must be truly random for the connection to be secure.
- The session key is then encrypted using the public key of the server.
- The encrypted session key is then sent to the server.
- The server decrypts the session key using its private key.

At this point, the transfer of a symmetric key to both parties has been accomplished. Both can now use this session key with symmetric key encryption for the remainder of the life of the connection.

For the server to be able to authenticate the client, several schemes could be used. The most obvious would be for the client to also have a certificate that is transmitted to the server on request. In practice, because of the expense of certificates, this is not routinely done. Within the WWW paradigm, the server returns a sign in page as the initial page served to any client once a secure connection is completed. As part of the sign in page, the server requests a user id and password. This is then submitted to the server across the secure connection. The server can then verify the user in much the same way as a login on a conventional time sharing computer system.

For the digital signature of a report or image we need a system that allows for access to a smartcard from within a WWW client. Two avenues are possible: A software plug-in could be written to send the report to the smartcard and get back the signature. Alternatively, we could use a JAVA applet and the newly defined smartcard application programmers interface (API)[12] to communicate with the smartcard. Once the digital signature is created, it is transported to the server and saved in the database along with the report. Later, when a report is accessed, the client can locally verify the digital signature of the report and thereby verify both the author and the report itself. If a timestamp of signatures is used, then one can further verify the time when the images and reports were created.

7. PERFORMANCE OF SECURE SYSTEMS

A Discussion of Security and Key Length

Whenever a cryptosystem is discussed issues of security under various forms of attack are also discussed. The security of the system must never rely on the secrecy of the algorithm. The security of an algorithm is therefore defined by the length of

time that it takes for an attacker to decrypt the ciphertext without knowledge of the key. The methods an attacker can use to decrypt a message can be varied.

There are some methods of encryption such as simple letter substitution algorithms that can be decrypted in a matter of seconds using a personal computer. Such algorithms are completely insecure and will not be considered here. For the most secure methods of symmetric key encryption, the best known method of attack is by brute force. A brute force attack will succeed when the attacker has one block of ciphertext where the corresponding plaintext is already known. This is a trivial requirement since most messages have a known header, for example Microsoft Word (Microsoft Corp, Redmond, WA) files, email messages, and so on. The attacker then decrypts this block of ciphertext with all possible keys until the known plaintext appears. As of November 1995, if the key is 40 bits or less, then a solution can on average be found in 2 seconds using $100,000 worth of custom hardware.[13] If a key length of 64 bits is used the average time to a solution is 37 days using $1 million worth of custom hardware. For systems that must be absolutely secure a key of 128 bits can be used, which would require 10^{11} years for a solution using $10 trillion dollars worth of custom hardware. For comparison purposes note that the national debt of the United States is $5 trillion dollars and that the estimated age of the universe is 10^{11} years. A secure algorithm with a 128 bit key can be considered absolutely secure for all purposes unless a method more efficient than brute force can be found and applied.

Another comparison between an attack's efficiency and key length is that of the factoring attack on RSA. The RSA algorithm relies on the difficulty of factoring a number n into two prime numbers p and q. At the moment no other attack on RSA will work faster than factoring n. The number n is part of the public key and is therefore known to everyone. If n is a 512 bit number it is estimated that it will take less than 200 years using a computer executing one million instructions per second to factor n. This number is actually changing because of improvement in the algorithms to factor large numbers. If n is a 1024 bit number it will take 3×10^7 mips-years to factor. This is an acceptable barrier unless you are signing a will or keeping a very long-lived secret.

All of these numbers were calculated using 1995 technology and are subject to Moore's law regarding the expansion of available computer power. Moore's law states that compute power doubles every 2 years. In some cases an algorithm may be broken, ie, some reasonable short cut to a solution can be found. This possibility is guarded against by using algorithms that have been in use for many years and have had attacks made on them by the best academic cryptographers.

The Impact of Security on System Performance

This section describes some results of a performance study on applying encryption in a networked medical imaging environment.[2] It was shown that when using a low speed transmission medium such as narrowband integrated services network (N-ISDN, 23 Kbits/s) that the encryption time for a large data set is 10% of the overall transmission time. At high speeds where the system response is nearly instantaneous (less than a second) the encryption time, which remains constant, is likely to be unacceptable. At Ethernet speeds, the overhead imposed is annoying but not egregious. Hardware based encryption should be used with higher network speeds. Such equipment has reasonable cost when compared to the costs of implementing high speed asynchronous transfer mode (ATM) networks. It was also found that by taking the time to perform lossless compression, there was a net gain in speed of transmission of 66%. Further studies should be done to compare the effects of compression on improving system performance versus the degradation caused by encryption.

8. CRYPTOGRAPHY AND THE LAW

The development of cryptographic algorithms for academic research or commercial applications presents significant problems to a spectrum of government agencies. Intelligence, law enforcement, and commerce are three facets of government that have diverse and sometimes conflicting positions with regard to applying cryptographic methods. Combine this with the public's concern for civil liberty and potential government encroachment and we have a vigorous national and international debate on security in the information society.

The United States government currently sponsors a number of initiatives to study and recommend appropriate policies for cryptography in general[14] and for healthcare in specific. One report was completed in August 1997 by the US Department of Health and Human Services (DHHS) Computer Sciences and Telecommunications Board (CSTB) of the National Research Council. The latter initiative has resulted in a comprehensive bibliography of publications relating to the confidentiality of electronic health data.[15] For an overview of the politics of cryptography in general see Schneier, chapter 25.[16]

Much of the current discussion of cryptography in medical records has been spurred by the US enactment of the Medical Records Confidentiality Act of 1995 (S1360). This law details the required mechanisms for handling of healthcare information and the remedies for violations of this Act. It charges the US Secretary of DHHS to establish regulations that safeguard medical record confidentiality. Such regulations are expected to be derived from the findings of the CSTB.

In practical terms, the application of cryptographic algorithms to medical records are subject to the laws and regulations in the country of use. The export of cryptographic algorithms is likewise heavily regulated in most countries. In the US, the government can consider cryptography a munition and it is listed on the US Munitions List. An unofficial nontreaty international organization Coordinating Committee for Multilateral Export Controls (CoCOM) was formed by the North Atlantic Treaty Organization (NATO) to coordinate national restrictions on various controlled technologies including cryptography. The regulations and restrictions go beyond the actual algorithms and include information required for the design, development, production, processing, manufacture, assembly, operation, repair, maintenance or modification of defense articles.[17]

In summary, the lawmaking and regulatory agencies of most governments have not kept pace with the diffusion of information technology and the need for secure communication in commercial enterprise. Governments are wrestling with the issues surrounding cryptographic policy, and medicine is one of many commercial applications that seeks clarity. Healthcare information system providers must therefore wrestle with a complex array of regulations on the development, sale, and export of cryptographic methods.

9. STANDARDS

The use of cryptography for commercial use is a relatively new phenomenon. Military applications have standards, but the military does not publish them. This is an attempt to increase security of military systems. Banking institutions have used the digital encryption standard (DES) symmetric key algorithm for commercial transactions together with the RSA public key algorithm since 1977. In recent years some banks have improved their symmetric key algorithm to triple-DES because of key length concerns. As the internet has grown in popularity, credit companies issuing Visa and Mastercard have joined forces to create the secure electronic transaction (SET) standard for the use of credit cards over the internet. There are also International Standards Organization of (ISO) standards for some aspects of cryptography. The ISO standard X.509[18] recommends RSA for public key cryptography and provides a template for certificates of public keys.

Governments are in various stages of acceptance of digital signatures, and cryptography in general, for legal purposes. In Japan, governmental committees will standardize and enforce digital security measures. In Germany, there is legal acceptance of digital signatures for contracts and affirmations. In the rest of the world the situation is less clear cut, but the overall direction seems clear.

Digital Imaging and Communications in Medicine (DICOM) Working Group 14, Security, in conjunction with CEN TC251 WG4, MEDIS-DC, and Japan Industries Association of Radiation Apparatus (JIRA) are working to add security specifications to the DICOM standard. There was general agreement that only one security standard be formed for all medical purposes. Initial proposals involve the use of the SSL protocol for the protection of privacy, and public key based digital signatures for making images and reports immutable. The following scenario indicates the scope of the problem and solutions for tracking the change history of images and reports that future releases of the standard should address.

This scenario is based on a computed tomographic (CT) study being the target of questioning during a court hearing. In this case it is necessary to prove that this particular data set was the set that

the radiologist or surgeon actually saw. The machine that made the scan would have first signed such a data set. This would tie the CT study to the manufacturer of the machine and possibly the particular machine used. The next signature would be from the radiologist, who would have signed the data set as well. The radiologist would also generate a report that included observations along with window, level, and any other adjustments used to view the data. Finally the surgeon would view the data and sign both it and his/her viewing adjustments. All of these signatures prove that a given physician or machine accessed the data set and that all of the bits are exactly as they were when he, she, or it signed the data set. So the signature renders the data set immutable.

In each case the signature should be sent to a Trent to timestamp the signature. The third party will not be able to access the data since all that is received is a signature which cannot be inverted to get at the data. This timestamp will only prove when the signature was received by the service. This prevents a radiologist or surgeon from changing the data or the report after the patient's results are less than satisfactory. Because the physicians can generate a legitimate signature at any time only a timestamp by a Trent can prove when the signature appeared. This concept of timestamping is not being considered for part of the current standard, but is being discussed for future releases.

Last, there must be a certificate system in place to insure that a key used to prove that a particular physician accessed data is inextricably bound to the physician. Certificates, in conjunction with digital signatures make signed images and documents nonreputable.

10. A WARNING

It may seem as if an article such as this one provides a clear blueprint of how to build and maintain secure systems. Although the mathematics, algorithms, and protocols described are the basis for many secure systems in use today, this knowledge is not enough. On June 19, 1996 the *New York Times*[19] published an article on the failure of a system thought to be secure. Mitsubishi Corp and Sumitomo Corp had formed subsidiaries to produce digital cash cards for a popular form of gambling in Japan called pachinko. The goal was to use an electronic cash system to ensure that taxes would be paid by this lucrative business.

The security of the electronic cash card system was based on three levels of protection. The cards themselves were of the magnetic stripe variety where the stripe was of a special material that would be difficult to manufacture. The data on the cards were encrypted so that no one could discern the data. Lastly, the cards required special writing machines to write the data. The system failed for a cost of $600 million.

The flaws in the system where as follows: The writing of cards only required a copy of the machines that were contained in every gambling outlet. These machines where simply stolen. The material did not have to be manufactured, the criminals simply used the old cards that where thrown away after they had been used up. Defeating the encryption was the easiest part; the encrypted data was simply copied from a known good card to old used cards. This is a well-known attack called block replay.[20] Clearly, the design and implementation of secure systems is a subtle art. No one should implement a secure system without the oversight of a competent person skilled in the art of cryptography.

11. CONCLUSION

In this article we have briefly explored some of the many facets of security with respect to medical information. Security, although a broad and complex field, cannot be ignored as legislation and regulations enforce its use. Also, the use of digital information in medicine continues to increase. Many tested algorithms, protocols, and tools exist with which to build secure systems, but their use should be by people well trained in the art of cryptography. The potential loss from a failed security implementation is high. Schneier[5] and Menezes[3] both wrote excellent books on security and provide a good starting place to learn more about cryptography.

REFERENCES

1. Lafrance S, Krok S, Moore R, et al: Security vs. Access: A New Health Care Dilemma, in Proceedings of the 1996 Annual HIMSS Conference. Chicago, IL, Healthcare Information and Management Systems Society, 1996, pp 1-9

2. Wong S: A Cryptologic Based Trust Center for Medical Images. JAMIA 3:410-421, 1996

3. Smith J: Authentication of Digital Medical Images with Digital Signature Technology. Radiology 194:771-774, 1995

4. McCurley K: Protecting Privacy and Information Integrity of Computerized Medical Information. http://www.cs.sandia.gov/~mccurly/health.html

5. Schneier B: Applied Cryptography (ed 2). New York, NY, John Wiley & Sons, 1996

6. Stinson D: Cryptography—Theory and Practice. Boca Raton, FL, CRC Press, 1995

7. Bach E, Bellouin S, Bernstein D: Cryptography-FAQ. http://www.cs.ruu.nl/wais/html/na-dir/cryptography-faq/.html

8. Menezes A, van Oorschot P, Vanstone S: Handbook of Applied CRYPTOGRAPHY, Boca Raton, FL, CRC Press, 1997, pp 9-10

9. Rivest R, Shamir A, Adleman L: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21(2):120-126, 1978

10. Schneier B: Applied Cryptography (ed 2). New York, NY, Wiley, 1996, pp 258

11. SSL 3.0 specification. Netscape, Sunnyvale, CA. http://www.jp.netscape.com/eng/ss13/, March 1996

12. JavaCard API Specification Version 1.0. Sun Microsystems Inc, Mountain View, CA. http://java.sun.com/products/commerce/doc.javacard.ps, October 1996

13. Schneier B: Applied Cryptography (ed 2). New York, NY, Wiley, 1996, pp 153

14. Dam K, Lin H: Cryptography's Role in Securing the Information Society. Washington, DC, National Academy Press, 1996. http://www2.nas.edu/cstbweb/28e2.html (prepublication copy ed.)

15. Auston I, Humphreys B, Clayton P: Confidentiality of electronic health data: methods for protecting personally identifiable information. Washington, DC, National Library of Medicine, US Department of Health and Human Services. http://www.nlm.nih.gov/pubs/cbm/confiden.html

16. Schneier B: Applied Cryptography (ed 2) New York, NY, Wiley, 1996, pp 597-618

17. US Department of State: International Traffic in Arms Regulations (ITAR), 22 CFR 120-130 (Office of Munitions Control, 1989)

18. Consultation Committee, International Telephone and Telegraph: Recommendation X.509, in The Directory-Authentication Framework. Geneva, Switzerland, International Telecommunications Union, 1989

19. Pollack A: Counterfeiters of a New Stripe Give Japan One More Worry: Fake Cards Thwart Efforts to End Pinball Scams. New York Times, Thursday June 20, 1996 (col. 2, pg. 1, sec. D)

20. Schneier B: Applied Cryptography (ed 2) New York, NY, Wiley, 1996, pp 191-193