Toward Managing Security Cost for Healthcare Information

Shihab A. Hameed, Habib Yuchoh
Department of Electrical and Computer Engineering,
Faculty of Engineering, IIUM University, KL, Malaysia,
E-mail: shihab@iium.edu.my

ABSTRACT- **Medical emergency and healthcare applications hold several sensitive data that requires suitable security techniques to prevent it. Data security, privacy and confidentiality are sample of security used for such applications. To hide those sensitive data from unauthorized accesses, the data need to be encrypted with key which is hold by authorized person. However, adding encryption to the data will increase the degradation in database performance due to: high processing time required of data encrypting and decrypting, the complexity of applying encryption to specific attributes of database, and the cost of key management. This research paper works toward managing the cost of having a suitable security level for sensitive healthcare data. It mainly based on considering multi-level of sensitivity for healthcare data and selecting the suitable encryption method for according to level of sensitivity and volume of data. This leads to eliminate the total processing time for encryption and decryption of whole data and consequently the total cost for security.**

Keywords- Cryptography, Medical Emergency System (MES), Security, Cost of Encryption

I. INTRODUCTION

MES is built to run on many types of hardware devices that use operating system, including laptops and or wearable computers. For ease of use, this allows flexibility when choosing a platform. MES is provides a graphical user interface that users can navigate with a stylus. . In the field, an emergency medical technician (EMT) would most likely use a handheld device to enter data; however, after the data has been loaded to a central database a web-based interface can be employed to allow an EMT or paramedic to view and edit the patient care report (PCR) on any internet enabled PC. The application is primarily menu driven, and data can be input by selecting preset information from database. There are drop-down menus that can be customizable drop-down minus to or by writing or typing in information. These will allow for quick navigation and data collection. Once the data has been up-loaded to a central database, a web-based interface can be used to edit the patient care report (PCR) on any internet enabled PC. The Medical Emergency Data Information is built using the client-server architecture. The application is hosted on a web server and responds to client requests. The client is usually a web browser. Each instance of a client can send requests to the application server. The server receives and responds to the requests. In order to fulfil a request, the application server might have to communicate with a database server. Client-server systems can be scaled horizontally and vertically.

A. Cryptography in HealthCare IT

In order to design secure health care IT, particularly the goal of patient self-determination must be translated into data objects, operations on data objects, rights and obligations for operations, etc. This has lead to various security models [1].

There are tens of possible encryption methods that could be used to secure medical data, but four are of particular importance. The Data Encryption Standard (DES) [2] is a candidate because of its pioneer status; it was long the standard by which all commercial encryptions were accomplished. The algorithm was implemented in hardware for even faster encryption and decryption. Triple-DES [3] is a candidate because it is the industry follow-on to DES. It is probably more secure, but two to three times slower than DES. The Advanced Encryption Standard (AES) [4] is the newest approved encryption standard. It operates with three different key sizes (128, 196, and 256 bits), and in some implementations runs faster than DES. RSA, the best known of the public cryptographic systems [5], has scalable security limited only by the chosen key size, but its performance is markedly slower than the other three symmetric key algorithms. Each of these algorithms offers something different, but our emphasis will be weighing security versus speed and its impact on workflow in a healthcare environment [6].

B. Health Level 7 Standards HL7

HL7, which is an abbreviation of Health Level Seven, is a standard for exchanging information between medical applications. This standard defines a format for the transmission of health-related information. HL7's initial involvement in the Health Insurance Portability and Accountability Act (HIPAA) legislation began in 1996 with the formation of the Attachments special interest group to standardize the supplemental information needed to support health care insurance. The HL7 version 3 standard has the aim to support any and all healthcare workflows. The v3 standard, as opposed to version 2, is based on a

978-0-7695-4959-0/13 $25.00 © 2013 IEEE
DOI 10.1109/ACSAT.2012.75

formal methodology (the HDF) and object oriented principles. [7]. Information sent using the HL7 standard is sent as a collection of one or more messages, each of which transmits one record or item of health-related information. Examples of HL7 messages include patient records, laboratory records and billing information. [8]

II. SECURITY SYSTEM

A. Password

Perl/ Javascript Secure User Authentication Project [9] implemented massage digest encryption scheme on both client and server machines to allow encrypted password protection for web-based Perl/CGI applications. Although there are many free Perl/CGI applications for password protection, only a few use SHA on the server-side, but the password still travels over the internet as plain text. All server-side only schemes (like .htaccess password protection) are completely open to packet-sniffing. With this scheme, the browser JavaScript encrypts the password on the client's machine, and session tracking allows only one response per session ID, making simple packet-sniffing and session replaying much more difficult.

B. Public Key Infrastructure

The Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke Digital Certificates. A Public Key Infrastructure (PKI) enables users of a basically unsecure public network such as the Internet to securely and privately exchange data through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority. [10]

C. Transparent data encryption

Transparent data encryption enables simple and easy encryption for sensitive data in columns without requiring users or applications to manage the encryption key. This freedom can be extremely important when addressing, for example, regulatory compliance issues. No need to use views to decrypt data, because the data is transparently decrypted once a user has passed necessary access control checks. Security administrators have the assurance that the data on disk is encrypted, yet handling encrypted data becomes transparent to applications. [11]

D. Three-Tier Architecture

A three-tier application architecture [12] provides a model for developers to create a flexible and reusable application. By breaking up an application into tiers (see Figure2), developers only have to modify or add a specific layer, rather than have to rewrite the entire application over. There should be a presentation tier, a business or data access tier, and a data tier. The Medical Emergency System was developed as a three-tier system. The first tier is house the database. The second tier is where the business logic and encryption server is located and the third tier is the client or user interface. Residing the business logic on a middle-tier allows for easy development, maintenance and deployment of the code. Using the three-tier architecture separates and makes applications easier to manage. Communication between the tiers is controlled. Therefore, changes made to one tier will not affect the other tiers. It provides effective distributed client-server architecture that increases performance, scalability and availability while hiding the complexity of program processing from the user.

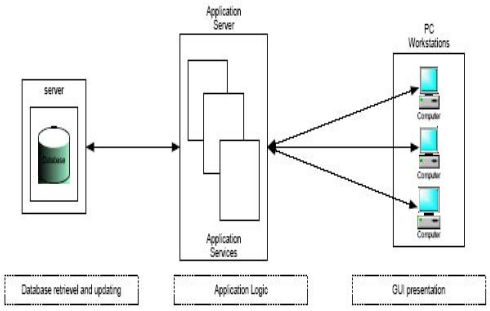

Figure2 Client-Server 3-tier Model

III. KEY MANAGEMENT

Key management deals with the secure generation, distribution, and storage of keys. Secure methods of key management are extremely important. Once a key is randomly generated, it must remain secret to avoid unfortunate mishaps (such as impersonation). In practice, most attacks on public-key systems will probably be aimed at the key management level, rather than at the cryptographic algorithm itself.

If someone's private key is lost or compromised, others must be made aware of this, so they will no longer encrypt messages under the invalid public key nor accept messages signed with the invalid private key. Users must be able to store their private keys securely, so no intruder can obtain them, yet the keys must be readily accessible for legitimate use. Keys need to be valid only until a specified expiration date but the expiration date must be chosen properly and publicized in an authenticated channel.

A. Session key

A session key is an encryption and decryption key that is randomly generated to ensure the security of a communications session between a user and another computer or between two computers. Session keys are sometimes called symmetric keys, because the same key is used for both encryption and decryption. Throughout each session, the key is transmitted along with each message and is encrypted with the recipient's public key. Because much of their security relies upon the brevity of their use, session keys are changed frequently. A different session key may be used for each message. [13]

B. *Public Key*

The public key of a pair can be known by anyone since there is no known way to deduce one key of a pair given the other. But it is critical to the security of messages encrypted by these algorithms that the corresponding private key of a pair be kept absolutely secret. The creation of these public/private pairs must be done with great care, must be effectively random and not predictable by an attacker, and must meet the requirements of the asymmetric key algorithm with which they are to be used. [14]

Public Key Generation Algorithm

1. Generate two large random primes, p and q, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.
2. Compute n = pq and (φ) phi = (p-1)(q-1).
3. Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1.
4. Compute the secret exponent d, 1 < d < phi, such that ed ≡ 1 (mod phi).
5. The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret.

C. *Key Equivalent*

Table 1 NIST Recommended Key Sizes

Symmetric Encryption	80	112	128	192	256
RSA (asymmetric encryption)	1024	2048	3072	7680	15380

Table 1 shows the key sizes recommended by the National Institute of Standards and Technology to protect keys used in conventional encryption algorithms like the (DES) and (AES) together with the key sizes for RSA that are needed to provide equivalent security[15][16].

IV. Encryption Cost Management

Encryption is a method or a process for protecting information from undesirable attacks by converting it into a non-recognizable form by its attackers. Data encryption mainly is the scrambling of the content of data, such as text, image, audio, video and so forth to make the data unreadable, invisible or incomprehensible during transmission. The goal is to protect the content of the data against the attackers. The reverse of data encryption is data decryption, which recovers the original data.

Asymmetric-key cryptography, also known as public-key cryptography, is a form of cryptography in which two digital "keys" are generated, one private and one public. These keys are used for encrypting or signing messages - one key is used to encrypt a message and another is used to decrypt it, or one key is used to sign a message and another is used to verify the signature. The public key can encrypt or sign messages that can only be verified using the private key, and vice-versa, so it is critical that the private key be kept secret.

Public-key algorithms can be used, depending on the protocol, for either confidentiality or sender authentication. For instance, a user can encrypt a message with their private key and send it. That it can be decrypted using the corresponding public key provides assurance that that user (and no other) sent it, unless the private key has been compromised. These algorithms can also be used for confidentiality; a message which is encrypted by the recipient's public key can only be decrypted by a person in possession of the paired private key.

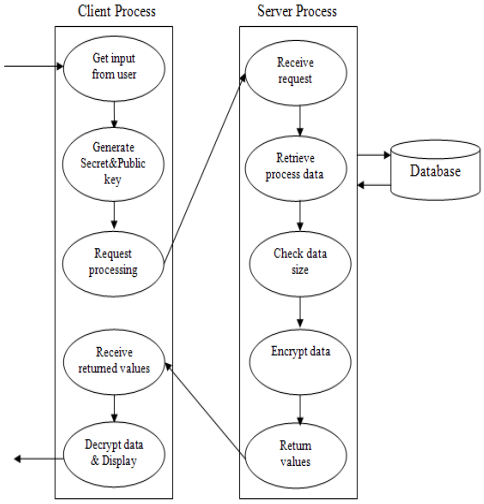

Figure 3 Inter Process Communication

Public key will be generated on the client system. The public portion of this key pair will reside on the servers being connected to; while the private portion needs to remain on a secure local area of the client system (see Figure3). In Figure 4, the key is generated and transmitted along with each message and encrypted with the recipient's public key. Because much of their security relies upon the brevity of their use, session keys are changed frequently. A different session key may be used for each message.

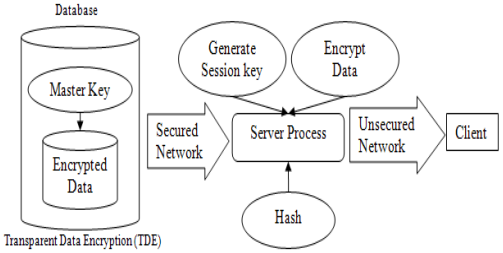

Figure4 Sever process as a middle tier

Public-key algorithms are very computationally costly, especially in comparison with many symmetric key algorithms of essentially equivalent security. This fact has important implications for practical use of these algorithms. To make a decision, we can use a factor of data size selecting appropriate algorithms to improve the process time of encryption operation (see Figure5 and Figure6).

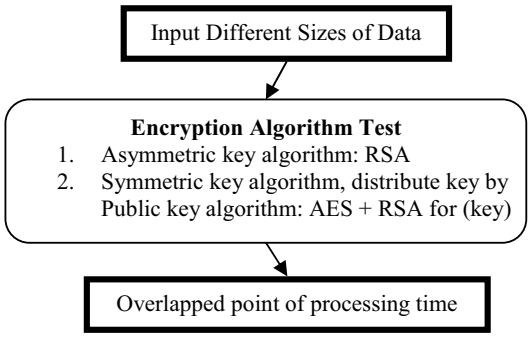

Figure 5 Candidate algorithm test with equivalent security

If someone's private key is lost or compromised, others must be made aware of this, so they will no longer encrypt messages under the invalid public key nor accept messages signed with the invalid private key. Users must be able to store their private keys securely, so no intruder can obtain them, yet the keys must be readily accessible for legitimate use. Keys need to be valid only until a specified expiration date but the expiration date must be chosen properly and publicized in an authenticated channel.

The middle tier will retrieve requested data from database and use data size checking to determine appropriate encryption algorithm. Encrypted data will be sent to the client and decrypt them into human readable form.

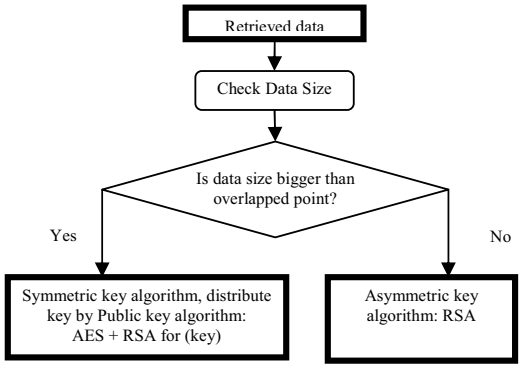

Figure6 Algorithm determinations from data size

V. Results And Discussion

The process of public key encryption (asymmetric key encryption) is relatively slow compared to private key encryption (symmetric key encryption) when working with large amount of data. However, it isn't all data size range that the processing time of asymmetric key encryption will be slower than symmetric key encryption. In some implementation, asymmetric key encryption can perform faster than symmetric key encryption (which includes key distribution) in some data size range especially in small amount data (see Table1).

Table 1 Comparing Encryption Cost

Message (Byte)	AES192 with RSA7683	RSA7683
2B	1446.6	41.3
4B	1446.7	38.4
6B	1446.6	324.7
8B	1446.6	344.3
10B	1446.6	663.0
12B	1446.7	663.7
14B	1446.7	954.0
16B	1446.7	878.7
18B	1446.6	1258.
20B	1446.7	1275.
22B	1446.8	1571.
24B	1446.6	1580.

The most effective way to speed up a system is to select suitable encryption algorithm by considering the data size. It can reduce encryption-process time and make system performs faster. This is a way to perform fast encryption and decryption process without reducing its security. However, there are some factors that should be considered. Firstly, key and values are chosen randomly. Secondly, secret key distribution uses public key. Thirdly, suitable tools and platforms involve the performance of healthcare services. Lastly, a good algorithm for high performance is also an effective method for solving a problem. Some of results may be different. It is depend on the factors which are unpredictable. However, this way is interesting direction that can reduce cost and process time.

VI. Conclusion

Hiding sensitive medical and healthcare data from unauthorized accesses requires encryption of data with key which is hold by authorized person. Encryption of data causes an additional cost for securing application.

Based on the suitable management model proposed in this paper, we found that the most effective way to speed up a system is to select suitable encryption algorithm by considering the data size. It can reduce encryption-process time and make system performs faster. This is a way to perform fast encryption and decryption process without reducing its security. The main factors that should be considered include: random choosing for key and values, are chosen randomly, using public key as secret key distribution, performance of healthcare services considered in selecting suitable tools and platforms, a good algorithm for high performance is also an effective method for solving a problem. However, this way is interesting direction that can reduce cost and process time.

REFERENCES

[1]. Joachim Biskup, Gerrit Bleumer. "Cryptographic protection of health information: cost and benefit". International Journal of Bio-Medical Computing 43 (1996) 61-67.

[2]. National Institute of Standards and Technology NIST, Computer security Resource Center, "Data Encryption Standard (DES)", FIPS 46-3,

http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

[3]. National Institute of Standards and Technology NIST, Computer security Resource Center, "Triple-DES (3DES)". (Retrieved Non 2009). http://csrc.nist.gov/cryptval. des.htm

[4]. National Institute of Standards and Technology NIST, "Advanced Encryption Standard," FIPS Publication 197, November 26, 2001, http://csrc.nist.gov/ publications/ fips/fips197/fips-197.pdf

[5]. RSA security Division of EMC, "RSA Security Inc". http://www.rsasecurity.com/ ,(Retrieved Nov 2009).

[6]. Snyder, A.M., and Weaver, A.C. "The e-Logistics of Securing Distributed Medical Data" IEEE International Conference on Industrial Informatics, Banff, Alberta, Canada, August 2003.

[7]. Health Level 7 International, "HL7". http://www.hl7.org, (Retrieved Nov 2010).

[8]. Health Level 7 International , "HL7 Overview",

www.interfaceware.com/manual/hl7.html, (Retrieved Nov 2010).

[9]. Perl Programming Language, Perl/Javascript MD5 Secure User Authentication. http://perl-md5-login.sourceforge.net/.

[10]. Omnisecu, "Public Key Infrastructure (PKI)".

http://www.omnisecu.com/security/public-key-infrastructure/what-is-public-key-infrastructure-pki.htm, (Retrieved Nov 2009).

[11]. Oracle, "Transparent Data Encryption". http://download.oracle.com/docs/cd/B19306_01/network.102/b14268/asotrans.htm, (Retrieved Nov 2009).

[12]. Kambalyal Channu. "3-Tier Architecture". http://channukambalyal.tripod.com/NTierArchitecture.pdf, (Retrieved Nov 2009).

[13]. David E. Marcinko .(2007). Dictionary of Health Information Technology and Security. Softcover, 448 pages. ISBN 0826149952

[14]. B. Kaliski, (1992). An Overview of Public-key Cryptography Standards. ConneXions, Volume 6, No. 5, May 1992

[15]. National security Agency NAS, The Case for Elliptic Curve Cryptography, http://www.nsa.gov/business/programs/elliptic_curve.shtml. (Retrieve Dec 2010),

[16]. Sandeep Sadanandan and Rajyalakshmi Mahalingam. (2008). Light Weight Cryptography and Applications. Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics 2008, 484-488, DOI: 10.1007/978-1-4020-8737-0_87