An enhanced security solution for electronic medical records based on AES hybrid technique with SOAP/XML and SHA-1

Article *in* Journal of Medical Systems · October 2013

DOI: 10.1007/s10916-013-9971-2 · Source: PubMed

CITATIONS
51

READS
419

4 authors:

Miss Laiha Mat Kiah
University of Malaya
116 PUBLICATIONS **3,711** CITATIONS
SEE PROFILE

Mohamed Abdulnabi
Asia Pacific University of Technology and Innovation
14 PUBLICATIONS **659** CITATIONS
SEE PROFILE

Bilal Bahaa
Universiti Pendidikan Sultan Idris (UPSI)
212 PUBLICATIONS **6,556** CITATIONS
SEE PROFILE

A. A. Zaidan
University Pendidikan Sultan Idris
225 PUBLICATIONS **6,732** CITATIONS
SEE PROFILE

J Med Syst (2013) 37:9971
DOI 10.1007/s10916-013-9971-2

ORIGINAL PAPER

An Enhanced Security Solution for Electronic Medical Records Based on AES Hybrid Technique with SOAP/XML and SHA-1

M. L. Mat Kiah · Mohamed S. Nabi · B. B. Zaidan · A. A. Zaidan

Received: 21 June 2013 / Accepted: 21 August 2013 / Published online: 14 September 2013

Abstract This study aims to provide security solutions for implementing electronic medical records (EMRs). E-Health organizations could utilize the proposed method and implement recommended solutions in medical/health systems. Majority of the required security features of EMRs were noted. The methods used were tested against each of these security features. In implementing the system, the combination that satisfied all of the security features of EMRs was selected. Secure implementation and management of EMRs facilitate the safeguarding of the confidentiality, integrity, and availability of e-health organization systems. Health practitioners, patients, and visitors can use the information system facilities safely and with confidence anytime and anywhere. After critically reviewing security and data transmission methods, a new hybrid method was proposed to be implemented on EMR systems. This method will enhance the robustness, security, and integration of EMR systems. The hybrid of simple object access protocol/extensible markup language (XML) with advanced encryption standard and secure hash algorithm version 1 has achieved the security requirements of an EMR system with the capability of integrating with other systems through the design of XML messages.

Keywords EMR security · RSA · AES · SHA-1 · SOAP · XML

M. L. M. Kiah · M. S. Nabi · B. B. Zaidan · A. A. Zaidan
Faculty of Computer Science and Information Technology, University Malaysia, 50603 Kuala Lumpur, Malaysia

B. B. Zaidan · A. A. Zaidan (✉)
Faculty of Engineering, Multimedia University, Jalan Multimedia, Cyberjaya 63100, Selangor Darul Ehsan, Malaysia
e-mail: aws.alaa@gmail.com

Introduction

The advancement of information technology has witnessed the automation and migration of many management systems previously handled manually, such as management of medical records. For accessibility and availability reasons, records are kept in digital format and are thus referred to as electronic medical records (EMRs). EMRs can be accessed and transmitted through different ways. A common way is through websites, which allows easy and convenient access to medical practitioners or patients. Given the nature of EMRs and the information they contain, these records should be kept and managed securely [1]. Ensuring the security of EMRs is one of the most important topics at present [2, 3], particularly when these records are being transmitted from one place to another. Figure 1 depicts the fluctuating number of publications related to EMR security. Generally, an increasing trend is apparent. The number of publications in international journals related to medical informatics has increased from 5.3 % in 2005 to 9 % in 2010. Ensuring the strong security of EMRs should not affect other requirements such as the scalability, integrity, and availability of these records [4].

Security requirements of EMRs

According to [1, 4], and [5], the security requirements for EMR involves authentication, authorization, integrity, non-repudiation, privacy, and confidentiality. Access to EMRs need must strictly be given only to authenticated individuals [1, 4, 5]. In handling these records, special permissions or rights should be exclusively granted to authorized individuals. The integrity of EMRs should be maintained [1, 6, 7] regardless of the transmission distance. Non-repudiation in EMRs should be achieved [4, 5, 8, 9] to ensure that no one can deny

Fig. 1 Papers per year discussing security issues related to EMRs in the international journal which related to the medical informatics (2005 – 2011)

receiving or sending the records. Privacy of EMRs relates to the right of individuals to keep their personal and private information from being exposed. Privacy of EMRs is required to be of the highest level [1, 10–12]. Confidentiality is one of the major issues concerning EMRs [1, 13, 14]. EMRs should always be highly confidential [1, 14–17], especially during transmission. Serious problems could occur if requested EMRs are not transmitted securely. For example, unauthorized modification of any record could result in serious repercussions in a patient diagnosis by a doctor. Thus, this study aims to ensure the secure transmission of EMRs. Several studies have focused on ensuring the security of EMRs. Furthermore, several encryption algorithms have been used to achieve the optimum secrecy of EMRs. However, complete solutions with high consideration of secrecy according to the functional requirements for EMRs are yet to be achieved.

Functional requirements of EMR systems

Similar with other online systems, an EMR system should be sufficiently scalable to accommodate the maximum number of users at a given time without degradation of its performance metrics [18] [19, 20]. An EMR system should be instantly available and operational whenever needed. EMR availability is considered a very important factor [19, 20]. In addition, the system should function promptly because the speed of information retrieval is very important in EMRs. Records need to be retrieved promptly to provide immediate care to patients [21, 22]. Integration of EMR systems should be considered as well. An EMR system requires to be integrated with other scattered systems when necessary [23, 24]. To date, only a few existing security solutions for EMRs that can fulfill the four main parameters of information security, namely, integrity, confidentiality, authentication, and non-repudiation, are available. Asymmetrical and symmetrical techniques of cryptography could provide greater security by using public and private keys to encrypt and decrypt messages. After critically reviewing existing algorithms and techniques, the present study proposes a hybrid technique that combines simple object access protocol/extensible markup language (SOAP/XML) and cryptography techniques to improve the security of EMRs.

Issues related to EMRs

EMR development started in the 1960s. The first EMR system was called COSTAR, which was developed at the laboratory of computer science at Massachusetts General Hospital [25]. Given the increasing importance and demand for EMRs, different issues related to EMRs correspondingly increase every year. Several international medical organizations started facing a range of issues with medical records. Security is one of the major issues concerning EMRs. Most medical journals have widely discussed issues related to EMRs. Figure 2 depicts the fluctuation in the number of publications focused on EMRs; nevertheless, generally, an increasing trend is apparent. The number of publications has increased from 10 % in 2005 to 16 % in 2010.

Literature survey

Numerous studies have been performed to ensure that EMRs are transmitted securely. All published studies are critically reviewed by the current paper and summarized in Table 1.

A total of 14 critical reviews were conducted. All of the studies discussed the issues of security; however, none of them could fulfill all the conditions, specifically the issue of confidentiality. Furthermore, majority of these studies did not fully achieve the security functionality of scalability, integration, and availability with XML in EMR transmission.

Methodology

The proposed research methodology consists of three phases. Phase one reviews and discusses EMR transmission and further examines the advantages and disadvantages of existing EMR security systems. Phase two proposes a new solution to enhance the security of EMR transmission. Finally, phase three evaluates and fine-tunes the proposed hybrid technique.

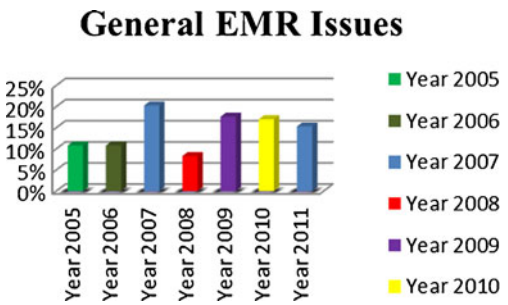


Fig. 2 Number of papers per year related to electronic medical records

Table 1 Critical review on EMRs

Source	Issues related to security in EMR	Security algorithms	Confidentiality	Use of XML in EMR Transmission	Security functionality		
					Scal	Integ	Ava
(Pekka	√	PKI	Not Achieved				
(Ishida & Sakamoto	√	RSA,DES & IDEA	Not Achieved				
(Gritzalis & Lambrinoudakis	√	Other	Not Achieved	√			
(Moehr & McDaniel	√	RSA	Not Achieved				
(Blobel & Holena	√	CORBA	Not Achieved				
(Schweiger	√	Other	Not Achieved	√		√	
(Snezana	√	RSA	Not Achieved			√	
(Masaya	√	Other	Not Achieved				√
(Stalidis	√	Other	Not Achieved	√		√	√
(Blobel	√		Not Achieved	√		√	√
(Lekkas	√	RSA	Not Achieved		√		√
(Papadakis	√	CORBA	Not Achieved	√			
(Blobel	√	Other	Not Achieved	√		√	
(Smith & Eloff	√	RSA	Not Achieved				√
(Ken	√	Other	Not Achieved				√
(Liu et al.	√	Other	Not Achieved	√		√	
(Rassinoux	√	Other	Not Achieved	√		√	

Possible solutions

This study aims to ensure the secure transmission of EMRs. By default, with public ports 80 (HTTP) and 443 (HTTPS) that are always open to allow dynamic content delivery and exchange, websites are constantly at risk of data theft and defacement. Meanwhile, EMRs cannot be sent between applications running on different operating systems with different technologies and programming languages without XML. Several researchers have proposed solutions to these problems; however, a better solution that considers the challenges of data security and integrity is yet to be proposed. Thus, this study proposes the use of SOAP/XML for exchanging information at the application layer. SOAP protocol structure over XML enables the integration among different service providers with a certain degree of security. To improve the level of security, two security schemes are proposed: (1) by adopting the algorithm by Rivest, Shamir, and Adleman (RSA algorithm); and (2) by adopting the advanced encryption standard (AES) algorithm. A hybrid of SOAP/XML and RSA with public key infrastructure (PKI) provides a certain level of security in services with integrity and non-repudiation. However, the RSA encryption algorithm is susceptible to different forms of security attacks and is therefore considered as a "broken algorithm." Given that the use of RSA algorithm is non-secure, an alternative solution using SOAP/XML with AES is proposed. The security of using these algorithms has been guaranteed. The integrity of messages and message authentication can be performed using hash codes. Secure hash algorithm version 1 (SHA-1) is the most widely used SHA hash function. It is presently employed in several security applications and protocols. Therefore, SOAP/XML with AES and SHA-1 security is the most secure system, given its low computation and bandwidth, applicability in any environment, scalability with any number of users, and accessibility at any time.

System development

In selecting and using a proper encryption algorithm to classify information from the viewpoint of both efficiency and security, [26] also reported that the cost performance of each algorithm should be considered to develop a practical system.

According to [1, 4–12, 14–17, 19–24], the system must ensure the security features, namely, confidentiality, integrity, authentication, non-repudiation, and resistance against quantum attacks. In addition, the system should guarantee interactive security functionality features, namely, scalability, availability, key exchanging, and integration. The overall goal of a system is to ensure and maintain 30 years of security, which is legally prescribed for EMRs [27, 28]. To achieve these requirements, the system is built using.NET technology of Microsoft visual studio, which acts as the platform for the

Table 2 RSA summary

The method	Privacy	Confidentiality	Integrity	Non-reputability	Authentication	Resistance against quantum attack	Scalability	Speed	Integration	Key exchanging	Availability
RSA	x	x	√	√	√	x	*	x	x	√	√

system. The web server is also built using .NET technology. Message exchanging is performed using SOAP/XML technology. Message exchange encryption is performed using AES algorithm with one-way encryption using SHA-1. Finally, the database used is Microsoft structured query language (SQL) server 2005.

RSA algorithm

RSA is an asymmetric algorithm. It is a public key cryptography that is sufficiently secure because of its long keys [29]. RSA involves three steps, namely, key generation, encryption, and decryption. RSA uses the concept of public and private keys. The public key is accessible to all users and is used in message encryption, whereas the private key is always kept secret and is used for message decryption. RSA is a deterministic encryption algorithm, that is, it has no random components. Therefore, it can be easily attacked by hackers and is thus considered non-secure. Furthermore, it cannot withstand quantum attacks [29]. Without proper processing of plain text, RSA encryption is fundamentally non-secure [30].

RSA is one of the most popular public key cryptographic algorithms. It is used in public signature applications and generally in secure transactions. It offers satisfactory cryptographic security; however, given its demanding mathematical calculation complexity, it is relatively slower than symmetric key algorithms [31]. As the security of RSA rests on unproven assumptions, this algorithm does not guarantee that the secrets it guards will remain safe for nearly 30 years [32]. According to Lomonaco (1999), if researchers succeed in building a feasible quantum computer, Shor's quantum factoring algorithm could break RSA easily. In the fall of 1999, a 512 bit RSA modulus was broken [33]. In RSA cryptography, the public key of each participant contains a large integer consisting of the product of two prime factors, called RSA modulus. If any of the two prime factors of a participant's public RSA modulus is found, the private key of that participant can be accessed, and the system is then considered broken. If these prime factors are sufficiently large, finding them is a complicated task [34]. Therefore, to ensure the security of the system, the primes must be kept sufficiently large. However, such measure leads to negative results. In fact, a large RSA modulus causes computational overheads under the RSA system. Thus, in RSA, a trade-off occurs between security and efficiency [33]. Integrity in RSA is maintained using message authentication codes (MAC), which are attached to each packet sent between stations over the network. In RSA, non-repudiation and message authentication can be achieved through digital signatures created by implementing a server public key authentication with clients [35]. Such step is performed by signing a unique message from the client with its private key. The signature is then returned to the client, who verifies it using the server's known public key [36]. Certificate-based authentication using 1024 bit RSA algorithm has sufficient scalability, whereas that using 2048 bit RSA algorithm has poor scalability [37]. The disadvantages of asymmetric authentication methods are that they are very time consuming and costly to implement in relation to hardware [38]. In general, PKI algorithms have limitations in overcoming different vendor problems. Furthermore, PKI is still unsuccessful in providing heterogeneous PKI standards for different vendors [39]. According to Wing and O'Higgins (1999), PKI has achieved operational availability to meet the constant expectations of its customers. Table 2 summarizes all the security features as well as the interactive security functionality features for RSA provided by the algorithms and methods discussed above. √ is the field value that indicates that the method or the algorithm supports the corresponding feature, whereas x denotes the opposite. In several special cases, the security feature is supported by the

Table 3 AES summary

The method	Privacy	Confidentiality	Integrity	Non-reputability	Authentication	Resistance against quantum attack	Scalability	Speed	Integration	Key exchanging	Availability
AES	√	√	*	**	√	√	√	√	√	x	√

Table 4 Hash value

Input number	Hashing algorithm	Hash value
16489	Input # X 197	3248333

algorithm or method when some enhancements are added. In such cases, * is used. Finally, ** indicates that the security feature is supported by the algorithm according to the author's conclusion.

AES algorithm

AES is a symmetric algorithm that has been adopted by the US government. It is extensively used worldwide. AES is publicly accessible and is the first cipher for top secret information approved by the National Security Agency (NSA). AES has a fixed block size of 128 bits and a fixed key size of 128, 192, or 256 bits. To break an AES 256 bit key, 2,200 operations are required, which takes longer than the age of the universe to complete. In other words, the AES algorithm is secure. Furthermore, this algorithm provides more physical security and higher speed [40]. Given the popularity of the AES algorithm, it is presently used as an authentication protocol [38]. The AES algorithm has been chosen as an encryption standard since 2001 and is considered to be extremely secure. Furthermore, it is suited for hardware implementations [38]. AES also provides data privacy [41]. To ensure the confidentiality of sensitive data, these must be ciphered with an encryption algorithm, such as AES [42]. Information security experts started imagining a threat to AES, that is, by quantum computers. A quantum computer has the ability to factor large numbers at an astonishing rate. Unlike regular computers, quantum computers can perform many calculations at once. Thus, AES should be backed up by newly created quantum algorithms to combat quantum attacks. Presently, a quantum computer that conducts transactions considerably faster compared with a normal computer is yet to be built [43]. According to Itani and Kayssi (2004), AES block cipher can provide data integrity. However, some researchers have criticized the ability of AES to provide data integrity without the help of other integrity-checking algorithms [41]. If the keys are exchanged securely between the receiver and the sender, and the encryption and decryption process is performed successfully, non-repudiation will certainly be attained in AES. Several studies have tested the scalability of AES algorithms and have proved that AES can achieve a very high throughput of 500 Gbits/s; thus, AES is extremely scalable [44]. AES provides exceptional scale integration, without requiring changes in the infrastructure or protocols [45–47]. Given that AES is highly efficient and can operate on a wide range of devices and processor types simultaneously [47], it is always available for the users' requests. Table 3 summarizes all security features as well as interactive security functionality features for AES provided by the algorithms and methods discussed above.

SHA-1

SHA-1 was designed by the NSA. Among the three SHA hash functions, namely, SHA-0, SHA-1, and SHA-2, SHA-1 is the most widely used. Hash algorithms are always used within other cryptographic algorithms and protocols to protect sensitive information. In a hash algorithm, encryption is based on a hash value. This value is calculated from a base input number using a hashing algorithm. Basically, the hash value is a summary of the original value. The interesting aspect about a hash value is that the original input number cannot be extracted without the data used to create the hash value. Table 4 depicts a simple example of the hash value.

Comprehending that the value 3,248,333 is the product of 16,489 and 197 is obviously difficult. However, if 197 is revealed as the multiplier, then the value 16,489 can be easily calculated. When a message is inserted into SHA-1, it produces an output called a message digest. This message can further be keyed into a signature algorithm that generates or verifies the signature for the message. The same hash algorithm as that used by the creator of the digital signature (SHA-1) must be used by the verifiers. A change in the message will result in a different message digest; thus, the signature will fail to verify. Therefore, SHA-1 is deemed secure because finding a message that corresponds to a given message digest or two different messages that produce the same message digest is computationally infeasible [48]. SHA-1 is only used for data integrity and message authentication [49, 50]. Scalability can be accomplished with the use of hash functions [51]. Hash function speed is dependent on the algorithm and application

Table 5 SHA-1 Summary

The method	Privacy	Confidentiality	Integrity	Non-reputability	Authentication	Resistance against quantum attack	Scalability	Speed	Integration	Key exchanging	Availability
SHA-1	x	x	√	x	√	x	√	√	√	√	√

Table 6 XML summary

The method	Privacy	Confidentiality	Integrity	Non-reputability	Authentication	Resistance against quantum attack	Scalability	Speed	Integration	Key exchanging	Availability
XML	x	x	*	*	*	x	√	*	√	√	√

complexity [52]. However, Michail, Kakarountas, Milidinis, and Goutis (2004) used SHA-1 to implement high-speed hash MAC. SHA-1 can be integrated with other algorithms, such as message digest 5 [53]. It can be used for implementing a key exchanging mechanism [54] and can achieve high throughput and operating frequency [55]. Table 5 summarizes all security features as well as interactive security functionality features for SHA-1 provided by the algorithms and methods discussed above.

XML

As discussed earlier, XML is presently considered as the universal language for data transmission or exchange over the Internet. XML has undergone several enhancements to improve its security and privacy features through XML encryption, XML signature, and XML key management specification (XKMS). XML encryption feature is added because it is considered as the best choice if the application requires a combination of secure and insecure communication, in which some of the data are exchanged securely while the rest are exchanged normally [56]. The purpose of XML Encryption is not to substitute secure socket layer and transport layer security protocols [56, 57]; instead, XML signatures add authentication, data integrity, and non-repudiation support to the data that they sign. The fundamental feature of XML signature is that it signs only precise parts of the XML tree rather than the entire document. As a result, other unsigned parts of the document can be changed by unauthorized people [58]. XKMS works simply as a web service that provides an interface between an XML application and a PKI. XKMS is one of the essential ways to ensure trust among users. It verifies the users and confirms whether they have the right to perform certain types of transactions. XML data are processed with great scalability [18, 59]. Data transfer speed rate over XML web services depends on numerous factors, such as Internet speed, compression of XML data, ancestor–descendant relationship between any pair of nodes in the hierarchy of XML data [60], and application of indices to speed up data transmission and retrieval [61]. Given that XML is platform and language independent, any XML-based solution is very flexible because it has the ability to integrate with any other system [62]. Flexibility, simplicity, and interconnection capabilities make XML an excellent language for data exchange over the Internet [63]. The reliability and availability of a web service can be achieved or improved with the use of SOAP and XML [64, 65]. Table 6 summarizes all security features as well as interactive security functionality features for XML provided by the algorithms and methods discussed above.

SOAP

SOAP is becoming a de facto standard because it is a lightweight protocol for exchanging structured and typed information [66]. SOAP is based on XML; therefore it can communicate through the Internet and is independent of the platform and programming language used. SOAP does not authorize an underlying transport protocol. Nevertheless, HTTP has been the most widely used protocol for SOAP [67]. In February 2001, Microsoft and IBM introduced SOAP header entry to carry digital signature information within SOAP 1.1 envelope to verify data origin and integrity [68]; however, a number of researchers have criticized SOAP. According to Brose (2003), SOAP does not provide message security; therefore, other ways of securing SOAP messages is necessary. Web services security assumes that message security should be determined at the web service definition language (WSDL) and SOAP levels [69, 70]. Many studies have attempted to increase the security model of SOAP, and research groups have expanded specifications for authentication, confidentiality, and

Table 7 SOAP summary

The method	Privacy	Confidentiality	Integrity	Non-reputability	Authentication	Resistance against quantum attack	Scalability	Speed	Integration	Key exchanging	Availability
SOAP	x	x	*	*	*	x	√	√	√	x	√

authorization using SOAP [69]. Thus, SOAP/XML can be used to send messages over the Internet with a certain level of security. However, the support of other security algorithms, such as AES, RSA, and SHA-1, is needed to achieve a high level of security. Shiping, Bo, Zic, Ren, and Ng (2006) mentioned that SOAP might not show ideal scalability when the amount of data transferred and the synchronizations increase. As that SOAP is a protocol that communicates using XML, it fully inherits the openness, scalability, and availability of XML [71]. According to Devaram and Anderson (2003), SOAP lags behind its peers in terms of speed. Hence, the idea of caching SOAP payloads at the client side was introduced to improve the overall speed and performance of SOAP. Similar to XML, SOAP assists in the integration of most applications [62]. Table 7 summarizes all security features as well as interactive security functionality features for SOAP provided by the algorithms and methods discussed above.

This new hybrid technique is proposed after a detailed discussion on the reasons behind selecting SOAP/XML web service with AES encryption. Features of different cryptography algorithms and data exchange languages were discussed and presented in table form for easy understanding. A critical analysis of the security features for algorithms and methods based on Tables 6 and 7 shows that SOAP/XML can guarantee scalability, speed, integration, key exchange, and availability during transmission, whereas AES with SHA-1 can guarantee security features in terms of providing data privacy, confidentiality, non-reputation, and authentication. To achieve the requirements of secure EMR transmission, this study proposes a hybrid technique using AES, SHA-1, and SOAP/XML. Table 8 summarizes the security features as well as interactive security functionality features provided by the algorithms and methods discussed above.

Based on the table above on cryptography algorithm, AES can guarantee data privacy, confidentiality, non-reputation (in special cases), scalability, high speed, fair integrity, integration, availability, and high resistance against quantum attacks. By contrast, RSA fails to guarantee the abovementioned security functions against quantum attacks. Nevertheless, unlike AES, RSA can provide integrity and key exchange. Meanwhile, SHA-1 algorithm can provide interactive security functionalities together with integrity and authentication.

Table 8 Summary of security features for algorithms/methods

Methods	Security						Interactivity of security functionality				
	Privacy	Confidentiality	Integrity	Non-reputability	Authentication	Resistance against quantum attack	Scalability	Speed	Integration	Key exchanging	Availability
RSA	x	x	√	√	√	x	*	x	x	√	√
AES	√	√	*	**	√	√	√	√	√	x	√
SHA-1	x	x	√	x	√	x	√	√	√	√	√
XML	x	x	*	*	*	x	√	*	√	√	√
SOAP	x	x	*	*	*	x	√	√	√	x	√
Proposed hybrids											
RSA & SHA-1	x	x	√	√	√	x	√	√	√	√	√
AES & SHA-1	√	√	√	**	√	√	√	√	√	√	√
SOAP/XML	x	x	*	*	*	x	√	√	√	√	√
RSA, SHA-1 & XML/SOAP	x	x	√	√	√	x	√	√	√	√	√
AES, SHA-1 & XML/SOAP	√	√	√	**	√	√	√	√	√	√	√

System flow

When the user (i.e., a doctor or a patient) sends the request message (Rm) to the server, several steps are involved. In Fig. 3, step 1 shows the message being encoded using XML and with SOAP before it is sent through HTTP. When the Rm is received by the server, the server decodes the message using SOAP and XML parser to obtain the requested values. Using these values, the web server generates an SQL query and

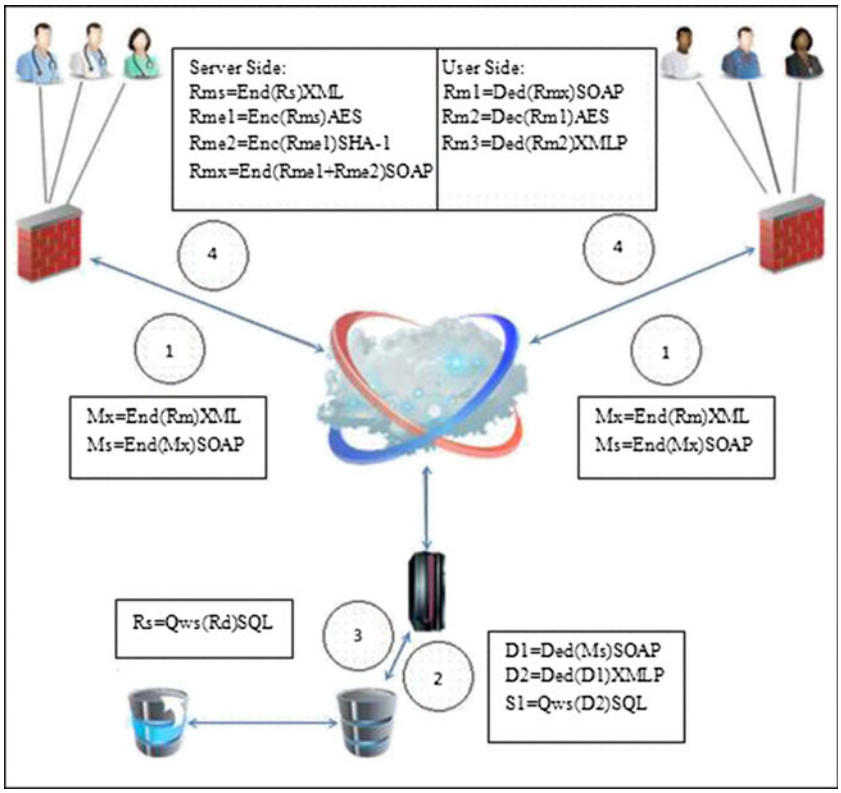


Fig. 3 System Flow

sends it to the database, as shown in Step 2. As a response, the database returns the result (Rs) of the query to the web server, as shown in Step 3. Once the server receives the result from the database, it encodes the message using XML and

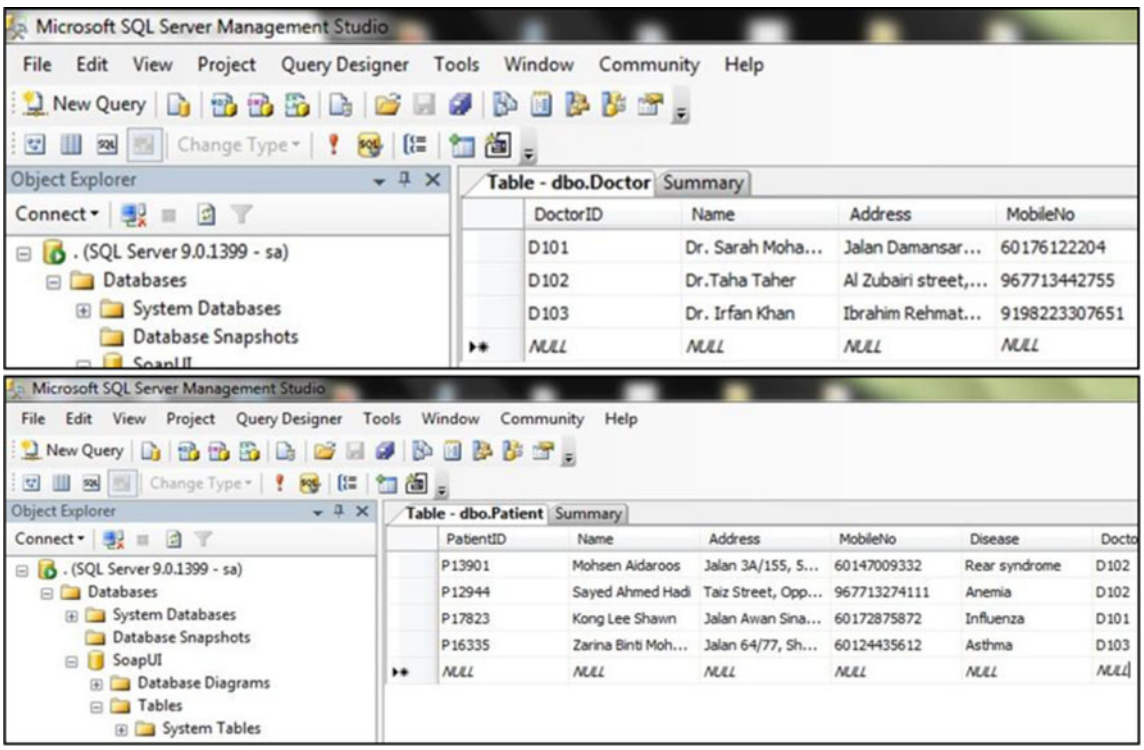


Fig. 4 Doctor and Patient details

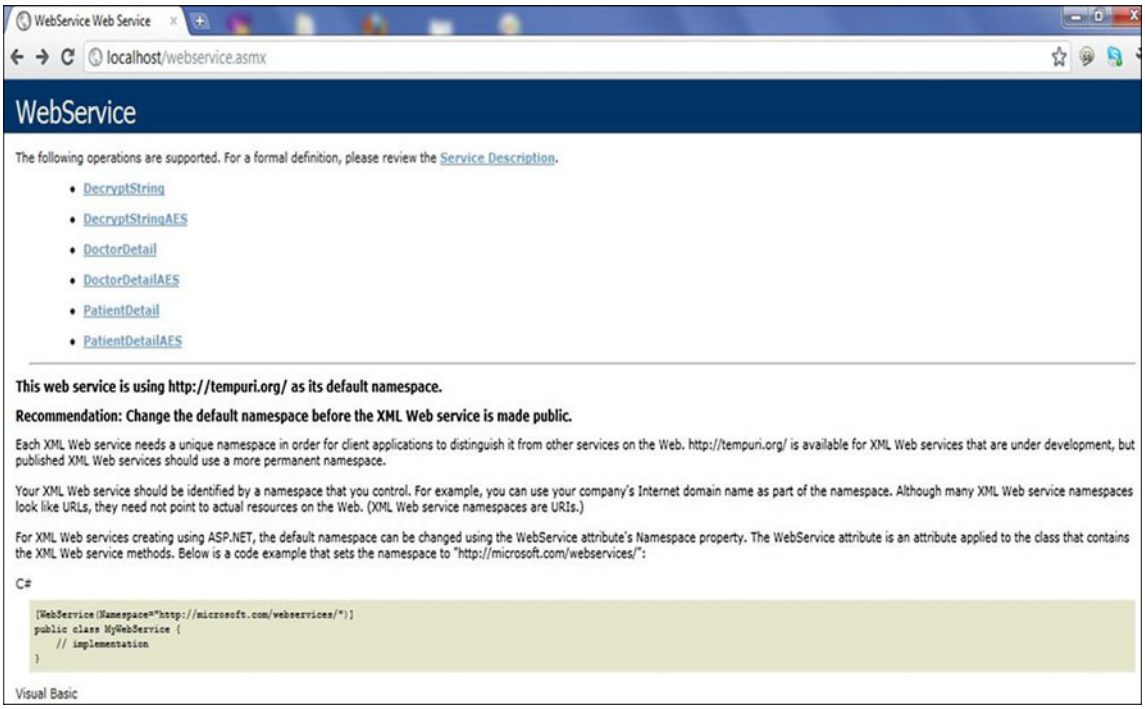


Fig. 5 Main screen

subsequently encrypts the message using AES algorithm. Here, the server applies one-way hash encryption using SHA-1. The web server then encodes the message using SOAP and sends it to the server, as depicted in Step 4. The details of the server side processes are also shown in Step 4. Upon receiving the encrypted message, the user will request the web server to decrypt the message. As a response, the web server will decode the message using SOAP and then decrypt it using AES algorithm. Lastly, the server will parse the message using XML parser. The details of the user side processes are also shown in Step 4.

System implementation

During implementation, AES encryption algorithm is used for encryption purposes to provide robust and resilient security. However, to test and compare AES and RSA, the system is also implemented with RSA. The following screenshots display the sample data added manually to the SQL server database management system to implement and test system performance. Figure 4 presents the doctor and patient details table in the SQL server database management system.

The main page appears when the following address is typed in the local browser. http://localhost/webservice.asmx.

As shown in Fig. 5, the main page has six options that provide service and links to a WSDL file.

DecryptString uses the RSA decryption algorithm to decrypt the string and obtain the original message.

DecryptStringAES uses the AES decryption algorithm to decrypt the string and obtain the original message. Once this option is clicked, the screenshot in Fig. 7a appears.

DoctorDetail uses the RSA encryption algorithm to obtain the requested doctor details.

DoctorDetailAES uses the AES encryption algorithm to obtain the requested doctor details. Upon clicking this option the screenshot in Fig. 7b appears.

PatientDetail uses the RSA encryption algorithm to obtain the requested patient details.

Table 9 Description of the main elements inside a WSDL file (Source: Christensen, Curbera, Meredith, & Weerawarna, 2001)

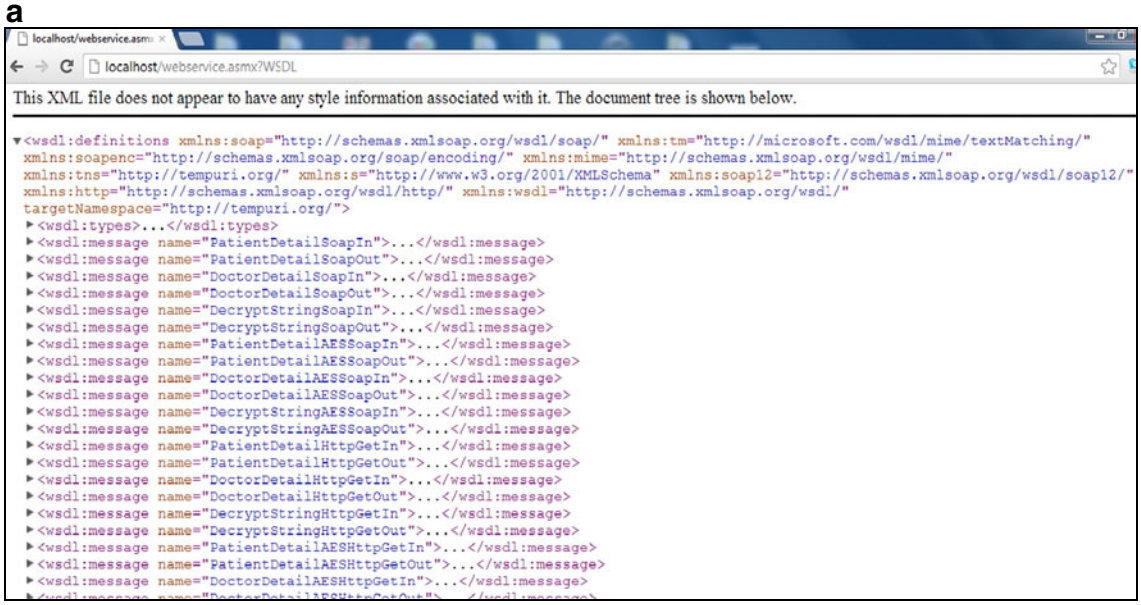
Element name	Description
Types	A container for data type definitions using some type system (such as XSD).
Message	An abstract, typed definition of the data being communicated.
Operation	An abstract description of an action supported by the service.
Port type	An abstract set of operations supported by one or more endpoints.
Binding	A concrete protocol and data format specification for a particular port type.
Port	A single endpoint defined as a combination of a binding and a network address.
Service	A collection of related endpoints

Table 10 Evaluation of the previous solutions with respect to the requirement of EMR

Security approach for medical records	Authors	Security						Interactivity of security functionality				
		Privacy	Confidentiality	Integrity	Non-reputability	Authentication	Resistance Against Quantum Attack	Scalability	Speed	Integration	Key Exchanging	Availability
The Benchmarks	(Rubio 2012) [84]	x	x	√	x	√	x	x	x	√	√	√
	(Cheng et al. 2010) [91]	x	x	x	x	x	x	x	x	√	x	√
	(Liu et al. 2008) [92]	x	x	x	x	x	x	x	x	√	x	x
	(Snezana 2007) [85]	x	x	x	x	√	x	x	x	√	x	x
	(Blobel et al. 2006) [86]	x	x	√	x	√	x	√	x	x	√	x
	(Schweiger et al. 2005) [93]	x	x	x	x	x	x	√	x	√	x	x
	(Pekka, 2004) [94]	x	x	x	x	x	x	x	x	√	x	x
	(Gritzalis and Lambrinoudakis 2004) [95]	x	x	√	x	√	x	x	x	x	x	x
	(Rassinoux et al. 2003) [96]	x	x	x	x	x	x	x	x	√	x	x
	(Lekkas et al. 2002) [87]	x	x	x	x	√	x	√	x	x	√	√
	(Stalidis et al. 2001) [97]	x	x	x	x	x	x	x	x	√	x	x
	(Papadakis et al. 2001) [98]	x	x	x	x	x	x	x	x	√	x	√
	(Blobel et al. 2001) [90]	x	x	x	x	x	x	x	x	x	x	x
	(Smith and Eloff 1999) [88]	x	x	x	x	√	x	x	x	√	x	√
	(Ishida and Sakamoto 1998) [26]	x	x	x	x	√	x	x	√	x	√	x
	(Moehr and McDaniel 1998) [89]	x	x	x	x	√	x	x	x	x	√	x
	(Masaya 1998) [99]	x	x	x	x	x	x	x	x	x	x	x
	(Ken 1998) [4]	x	x	x	x	x	x	x	x	x	x	x
Our Propose		√	√	√	√	√	√	√	√	√	√	√

(Please note: √ refers to achieved; x refers to not achieved)

a



b

Fig. 6 **a:** WSDL file **b**: WSDL file

PatientDetailAES uses the AES encryption algorithm to obtain the requested patient details.

Service description

WSDL, which is usually in XML format, describes the network services, operations, and means to access them. It acts as an interface for the web service. The definition and description of the standard elements inside a WSDL file are shown in Tables 9 and 10.

Figure 6a and b show screenshots of the WSDL file taken from the system by clicking on the service description link.

Figure 7a shows that the AES encrypted string needs to be copied and then placed in the text box area. After the encrypted string is pasted, the invoke button should be clicked.

Figure 7b shows that doctor details can be retrieved using the AES encryption algorithm. The details can be retrieved once the doctor's ID is keyed-in inside the text box area.

Whenever any option from the main screen is clicked, a sample SOAP request and response as well as HTTP GET and HTTP response messages are automatically generated by the web server and are displayed on the screen. To avoid duplication, only the DoctorDetailsAES option from the main

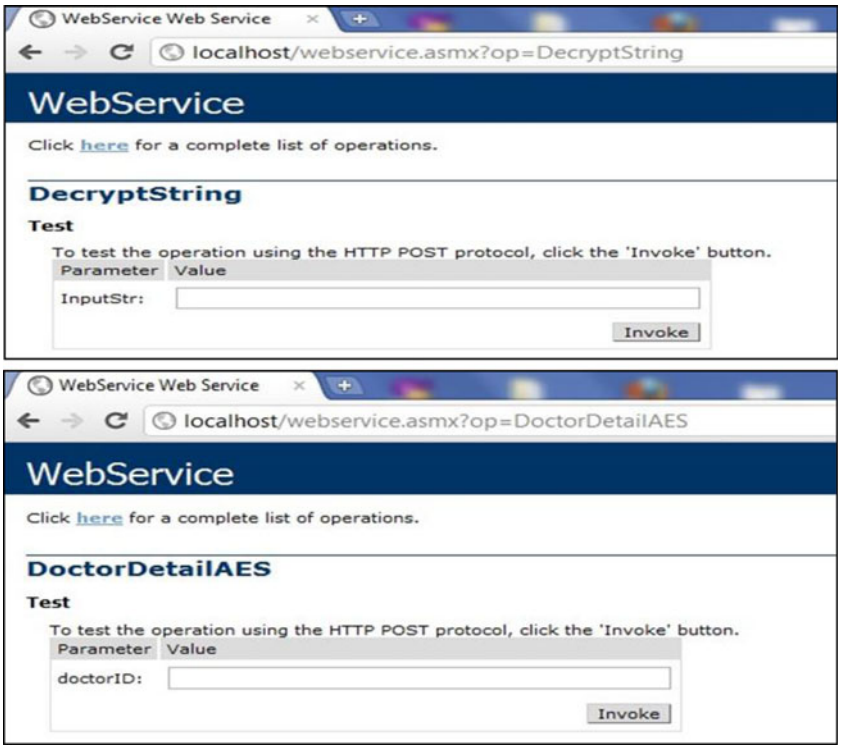


Fig. 7 **a**: Decrypt string using AES **b**: Doctor Details using AES

screen was selected to show these messages. Figure 8a and b show the sample SOAP 1.1 and 1.2 request and response messages generated by the web service.

Figure 8c shows the sample HTTP GET and HTTP POST request and response generated by the web server when the DoctorDetailAES option is clicked in the main screen.

System testing

The module was tested to ensure that all individual components within the module are working properly according to the requirements. Moreover, testing was performed to guarantee that the results of the test meet the expectations and that the system is functional and works efficiently without any errors or bugs. The proposed module was tested using a "white box" testing method. A white box test basically depends on the internal structure of the software. It is derived by executing specific elements of the code [72]. This test can be used either to produce test cases or to appraise the degree to which a given set of test cases covers all the code elements identified by the approach [72].

White box testing for web applications

Tonella and Ricca (2004) explained that white box testing methods are usually performed to test web applications. Such tests employ two models, namely, navigation and control flow models. The first model describes the user interaction in terms of pages visited, links followed, and forms submitted, whereas the second model tests the execution of the individual instructions at the server or client side.

Navigation model testing

The navigation model describes the web application related to composing pages and allowed links. In this model, both dynamic and static pages are tested. Dynamic pages are the result of the execution program on the web server in response to a request from the web browser, whereas static pages contain fixed information [73, 74]. Dynamic web pages depend on the information provided by the user through the input fields. The following tests were conducted in line with the navigation model using the white box testing method.

Test case: Encryption models

Five types of test cases were implemented to ensure that the encryption works without any glitches. Encryption testing was carried out on doctor details, and similar tests were performed on patient details, as follows.

a) Testing Doctor Details using AES

Doctor details were tested in this test case. As shown in Fig. 9a, the doctor with the ID "D101" was selected to perform this test. Once the doctor's ID (i.e., "D101") is entered and the invoke button is pressed, the information about the doctor appears in encrypted format using AES encryption method, as depicted in Fig. 9b.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /webservice.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/DoctorDetailAES"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DoctorDetailAES xmlns="http://tempuri.org/">
      <doctorID>string</doctorID>
    </DoctorDetailAES>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DoctorDetailAESResponse xmlns="http://tempuri.org/">
      <DoctorDetailAESResult>xmlxml</DoctorDetailAESResult>
    </DoctorDetailAESResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /webservice.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <DoctorDetailAES xmlns="http://tempuri.org/">
      <doctorID>string</doctorID>
    </DoctorDetailAES>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <DoctorDetailAESResponse xmlns="http://tempuri.org/">
      <DoctorDetailAESResult>xmlxml</DoctorDetailAESResult>
    </DoctorDetailAESResponse>
  </soap12:Body>
</soap12:Envelope>
```

HTTP GET

The following is a sample HTTP GET request and response. The placeholders shown need to be replaced with actual values.

```
GET /webservice.asmx/DoctorDetailAES?doctorID=string HTTP/1.1
Host: localhost
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<DataTable xmlns="http://tempuri.org/">xmlxml</DataTable>
```

HTTP POST

The following is a sample HTTP POST request and response. The placeholders shown need to be replaced with actual values.

```
POST /webservice.asmx/DoctorDetailAES HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

doctorID=string
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<DataTable xmlns="http://tempuri.org/">xmlxml</DataTable>
```

Fig. 8 **a**: SOAP 1.1 request and response **b**: SOAP 1.2 request and response **c**: HTTP GET & HTTP POST request and response

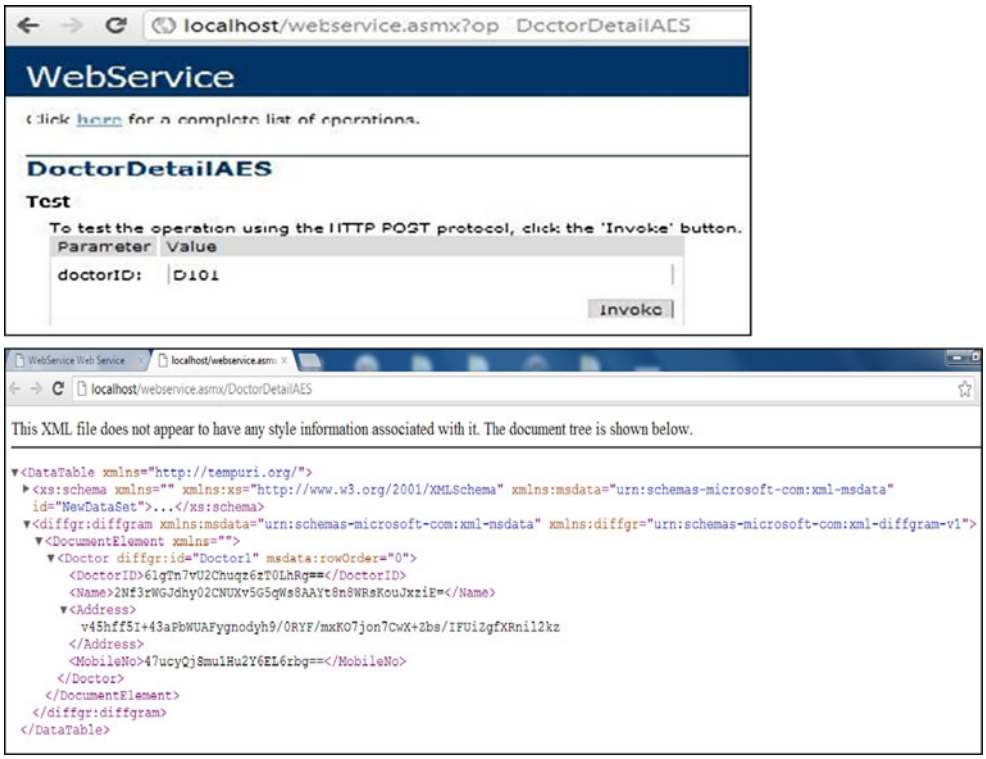


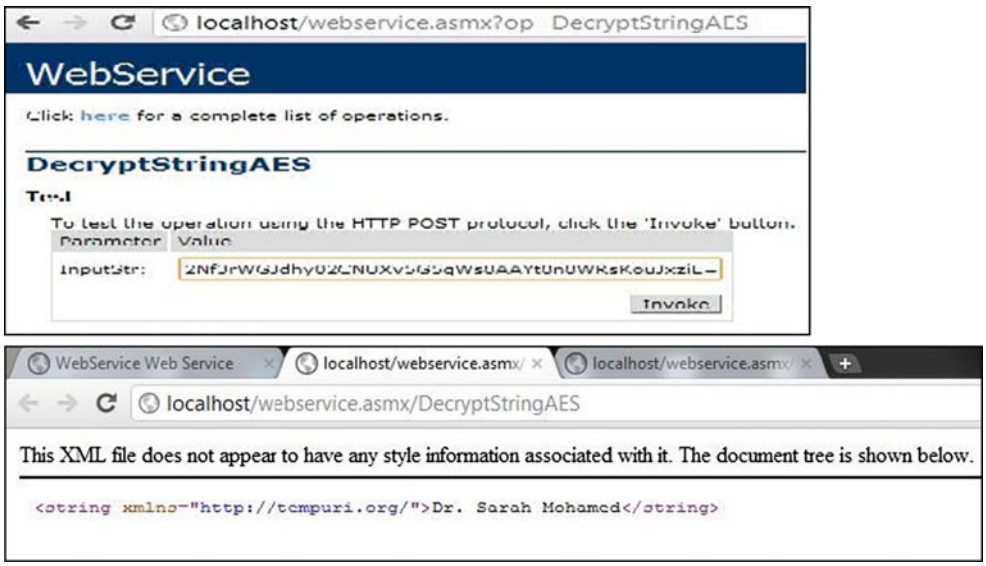
Figure 9b shows that each individual field (i.e., doctor's ID, name, address, and mobile phone number) is encrypted using AES algorithm.

Once the encrypted message is generated, it needs to be decrypted to obtain the original message. For the purpose of testing, name and address were selected to be decrypted (Fig. 10a and b).

The encrypted doctor's name string needs to be copied and then pasted in the input string text box area, as shown in Fig. 10a. Once the invoke button is pressed, the string is decrypted and the original message appears, as shown in Fig. 10b.

System evaluation

In order to evaluate the system, table based evaluation method has been adopted [75–83]. Although RSA has been widely used in securing EMR, this study proved that RSA is a broken algorithm. According to [26, 84–88], and [89], RSA



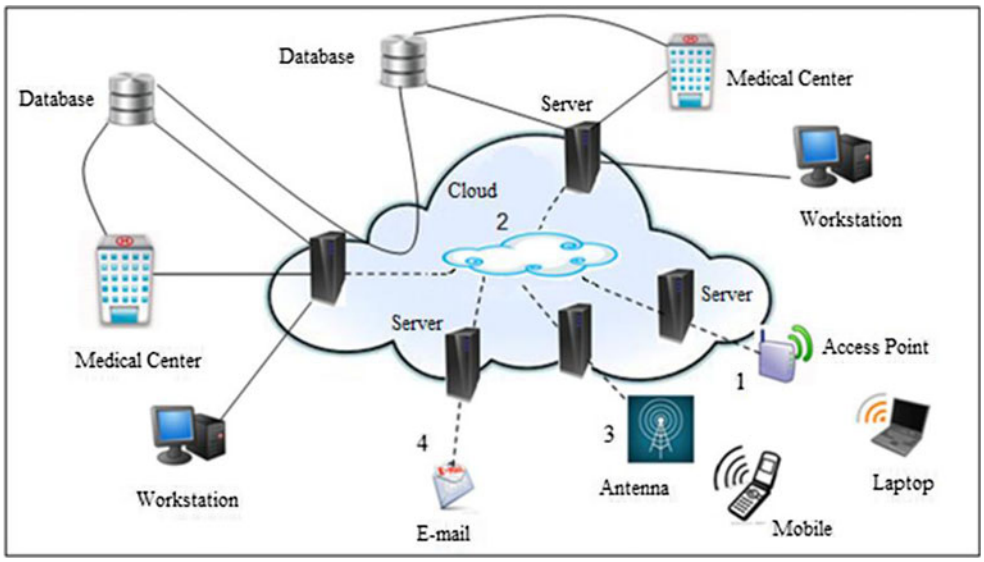


Fig. 11 Illustration of future work

encryption algorithm has been used and proposed for securing EMRs. These solutions were therefore considered as non-secure; thus, privacy and confidentiality have not been achieved. By contrast, [90] claimed that privacy and confidentiality for EMR have been achieved; however, no technical details about the encryption algorithm used in the implementation are available. [91–99], and [4] do not provide a proper evaluation for the technique to be used. Therefore, these studies cannot be considered for building a secured platform for EMRs.

The proposed approach was built on XML/SOAP, which provides integration, scalability, availability, fast data transmission, and key exchange. The encryption process was handled by AES and hash algorithms to provide privacy, confidentiality, non-repudiation, integrity, authentication, and resistance against quantum attacks. Thus, the solution in this study provides all the features required in a system for transmitting EMRs.

Conclusion

In conclusion, the proposed SOAP/XML hybrid technique with AES and SHA-1 achieved the objectives listed in the literature in terms of securing EMRs. The hybrid technique obtained top confidentiality with non-repudiation for EMRs. Aside from the main security features, the technique also attained the other important features, such as scalability, availability and integration. The possibility of using an asymmetric encryption algorithm, such as RSA, was also studied and implemented. RSA encryption algorithm can be implemented and can provide a viable mechanism for key exchanging. However, it failed to achieve the high level of confidentiality required for EMR transmission, given that RSA is a broken algorithm.

Summary points

What was already known:

- EMRs need to be kept secure.
- The use of XML in EMR transmission allows efficient transmission throughout heterogeneous systems.
- Security needs to be maintained during EMR transmission.

What this study added:

- Actual security requirements for transmitting EMRs were identified.
- A secure framework for transmitting EMRs was proposed.
- An overview of the issues to be considered to ensure the security in EMR transmission is presented.

Future directions

A study cannot be considered complete unless it covers all the possible areas of implementation. However, this task cannot be achieved easily because every researcher is made to draw a research boundary due to time constraint and degree requirements. In several cases, other enhancements can be introduced to the research to achieve additional goals. Nevertheless, such enhancements were not included in this study due to time constraint. Figure 11 shows the areas covered and those that are not included in this study, which are represented by straight and dotted lines, respectively.

Acknowledgments This Research has been partially funded from high impact research unite (HIR) at University of Malaya, under grant number UM.C/HIR/MOHE/FCSIT/12. A special thank goes to Multimedia University and Sunway University for providing several researches facilities, important recourses and providing experts consultations to improve this work.

References

1. van der Linden, H., Kalra, D., Hasman, A., and Talmon, J., Inter-organizational future proof EHR systems: A review of the security and privacy related issues. *Int. J. Med. Inform.* 78(3):141–160, 2009.
2. David, T., Securing access to healthcare. *Biom Technol Today* 2011(2):10–11, 2011.
3. Krawczyk, S., and Jain, A. *Securing electronic medical records using biometric authentication*. Springer, 2005.
4. Toyoda, K., Standardization and security for the EMR. *Int. J. Med. Inform.* 48(1–3):57–60, 1998.
5. Ruotsalainen, P., and Manning, B., A notary archive model for secure preservation and distribution of electrically signed patient documents. *Int. J. Med. Inform.* 76(5–6):449–453, 2007.
6. Chang, I. C., Li, Y.-C., Wu, T.-Y., and Yen, D. C., Electronic medical record quality and its impact on user satisfaction — Healthcare providers' point of view. *Gov. Inf. Q.* 29(2):235–242, 2012.
7. Beahan, S., In: Thomas, P. (Ed.), *10 - Legal Issues in Medical Records/Health Information Management, in Practical Guide to Clinical Computing Systems*. Academic Press, New York, pp. 171–180, 2008.
8. Ting, D., Securing access to healthcare. *Biom Technol Today* 2011(2):10–11, 2011.
9. Lekkas, D., and Gritzalis, D., Long-term verifiability of the electronic healthcare records' authenticity. *Int. J. Med. Inform.* 76(5–6):442–448, 2007.
10. Perera, G., Holbrook, A., Thabane, L., Foster, G., and Willison, D. J., Views on health information sharing and privacy from primary care practices using electronic medical records. *Int. J. Med. Inform.* 80(2): 94–101, 2011.
11. Yang, C.-M., Lin, H.-C., Chang, P., and Jian, W.-S., Taiwan's perspective on electronic medical records' security and privacy protection: Lessons learned from HIPAA. *Comput. Methods Prog. Biomed.* 82(3):277–282, 2006.
12. Peleg, M., Beimel, D., Dori, D., and Denekamp, Y., Situation-based access control: Privacy management via modeling of patient data access scenarios. *J. Biomed. Inform.* 41(6):1028–1040, 2008.
13. Kurtz, G., EMR confidentiality and information security. *J. Healthc. Inf. Manag.: JHIM* 17(3):41–48, 2003.
14. Barrows, R. C., and Clayton, P. D., Privacy, confidentiality, and electronic medical records. *J. Am. Med. Inf. Assoc.* 3(2):139–148, 1996.
15. Likourezos, A., Chalfin, D. B., Murphy, D. G., Sommer, B., Darcy, K., and Davidson, S. J., Physician and nurse satisfaction with an Electronic Medical Record system. *J. Emerg. Med.* 27(4):419–424, 2004.
16. Lim, E. Y. S., In: David Dagan, F. (Ed.), *11 - Data Security and Protection for Medical Images, in Biomedical Information Technology*. Academic Press, Burlington, pp. 249–257, 2008.
17. Mohan, J., and Razali Raja Yaacob, R., The Malaysian Telehealth Flagship Application: A national approach to health data protection and utilisation and consumer rights. *Int. J. Med. Inform.* 73(3):217–227, 2004.
18. Litoiu, M., *Migrating to Web services - latency and scalability*. in *Web Site Evolution, 2002. Proceedings. Fourth International Workshop on*. 2002.
19. Van de Velde, R., Framework for a clinical information system. *Int. J. Med. Inform.* 57(1):57–72, 2000.
20. Xue, Y., Liang, H., Wu, X., Gong, H., Li, B., and Zhang, Y., Effects of electronic medical record in a Chinese hospital: A time series study. *Int. J. Med. Inform.* 81(10):683–689, 2012.
21. Zimmerman, T. G., The case for electronic medical records—why the time to act is now. *Osteopath. Fam. Physician* 2(4):108–113, 2010.
22. Lucas, L., Partnering to enhance the nursing curriculum: Electronic medical record accessibility. *Clin. Simul. Nurs.* 6(3):e97–e102, 2010.
23. Rose, A. F., Schnipper, J. L., Park, E. R., Poon, E. G., Li, Q., and Middleton, B., Using qualitative studies to improve the usability of an EMR. *J. Biomed. Inform.* 38(1):51–60, 2005.
24. Hannan, T. J., Variation in health care—the roles of the electronic medical record. *Int. J. Med. Inform.* 54(2):127–136, 1999.
25. Mandl, K. D., Szolovits, P., Kohane, I. S., Markwell, D., and MacDonald, R., Public standards and patients' control: how to keep electronic medical records accessible but privateMedical information: access and privacyDoctrines for developing electronic medical recordsDesirable characteristics of electronic medical recordsChallenges and limitations for electronic medical recordsConclusionsCommentary: Open approaches to electronic patient recordsCommentary: A patient's viewpoint. *Bmj* 322(7281):283–287, 2001.
26. Ishida, Y., and Sakamoto, N., A secure model for communication of health care information by sub-division of information and multiplication of communication paths. *Int. J. Med. Inform.* 49(1):75, 1998.
27. Brandner, R., Van der Haak, M., Hartmann, M., Haux, R., and Schmucker, P., Electronic signature of medical documents-integration and evaluation of a public key infrastructure in hospitals. *Methods Inf. Med.* 41(4):321–330, 2002.
28. Beyer, A., Hellmann, S., Hesse, M., Holl, F. L., Morcinek, P., Paulus, S., Reimer, H., Dahms, M., Kausmann, K., and Friedrich-Meier S., *Criteria for success of identification, authentication and signing methods based on asymmetric cryptographic algorithms (EKIAS).* 2007.
29. Gottesman, D. and Lo, H. K., *From quantum cheating to quantum security.* Arxiv preprint quant-ph/0111100, 2001.
30. Boneh, D., Joux, A., and Nguyen, P. Q., *Why Textbook ElGamal and RSA Encryption Are Insecure*, in *Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, Springer-Verlag. p. 30–43,2000.
31. Fournaris, A. P., and Koufopavlou, O., *A new RSA encryption architecture and hardware implementation based on optimized Montgomery multiplication*. in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*. 2005.
32. Robinson, S., Still guarding secrets after years of attacks, rsa earns accolades for its founders. *SIAM News* 36(5):1–4, 2003.
33. Lenstra, A. K., *Recent developments in cryptography.* Information Security Summit, p. 30–31, 2001.
34. Lenstra, A. K., and Verheul, E. R., Selecting cryptographic key sizes. *J. Cryptol.* 14(4):255–293, 2001.
35. Ganesan, R., *Yaksha: augmenting Kerberos with public key cryptography*. in *Network and Distributed System Security, 1995., Proceedings of the Symposium on*. 1995.
36. Pellegrini, A., Bertacco, V., and Austin, T., *Fault-based attack of RSA authentication*. 2010.
37. Song, R., and Korba, L., *Scalability of Security Technologies on Multi-agent Applications*, 2003.
38. Feldhofer, M., Dominikus, S., and Wolkerstorfer, J., Strong authentication for RFID systems using the AES algorithm. *Cryptogr. Hardw. Embed. Syst.-CHES* 2004:85–140, 2004.
39. Medani, A., Gani, A., Zakaria, O., Zaidan, A., and Zaidan, B., Review of mobile short message service security issues and techniques towards the solution. *Sci. Res. Essays* 6(6):1147–1165, 2011.
40. Xinmiao, Z., and Parhi, K. K., High-speed VLSI architectures for the AES algorithm. Very Large Scale Integration (VLSI) Systems. *IEEE Trans.* 12(9):957–967, 2004.
41. Elbaz, R., Torres, L., Sassatelli, G., Guillemin, P., and Bardouillet, M., *PE-ICE: Parallelized Encryption and Integrity Checking Engine*. in *Design and Diagnostics of Electronic Circuits and systems, 2006 IEEE*. 2006.
42. Vaslin, R., Gogniat, G., Diguet, J.-P., Tessier, R., and Burleson, W., *Low latency solution for confidentiality and integrity checking in embedded systems with off-chip memory*. in *ReCoSoc proceeedings 2007*. 2007.

43. Asenjo, J. C., The Advanced Encryption Standard–Implementation and Transition to a New Cryptographic Benchmark. *Netw. Secur.* 2002(7):7–9, 2002.
44. Bouhraoua, A., *Design Feasibility Study For A 500 Gbits/s AES Cypher Decypher Engine*. in *Microelectronics, 2006. ICM '06. International Conference on*. 2006.
45. Shen-Fu, H., Ming-Chih, C., and Chia-Shin, T., Memory-free low-cost designs of advanced encryption standard using common subexpression elimination for subfunctions in transformations. Circuits and Systems I: Regular Papers. *IEEE Trans.* 53(3):615–626, 2006.
46. Chih-Chung, L. and Shau-Yin, T., *Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter*. in *Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on*. 2002.
47. Itani, W., and Kayssi, A., J2ME application-layer end-to-end security for m-commerce. *J. Netw. Comput. Appl.* 27(1):13–32, 2004.
48. Eastlake, D. and Jones, P., *Network Working Group D. Eastlake, 3rd Request for Comments: 3174 Motorola Category: Informational P. Jones Cisco Systems September 2001.* 2001. **RFC 3174.**
49. Quinlan, S. and Dorward, S., *Venti: a new approach to archival storage*. 2002.
50. Madson, C. and Glenn, R., *The use of HMAC-MD5-96 within ESP and AH.* 1998.
51. Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H., Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.* 31(4):149–160, 2001.
52. Xiao, D., Liao, X., and Deng, S., One-way Hash function construction based on the chaotic map with changeable-parameter. *Chaos, Solitons Fractals* 24(1):65–71, 2005.
53. Mao-Yin, W., Chih-Pin, S., Chih-Tsun, H., and Cheng-Wen, W., *An HMAC processor with integrated SHA-1 and MD5 algorithms*. in *Design Automation Conference, 2004. Proceedings of the ASP-DAC 2004. Asia and South Pacific*. 2004.
54. Traw, C. B. S., and Aucsmith, D. W., *Content protection for transmission systems*, Google Patents, 1999.
55. Michail, H. E., Kakarountas, A. P., Milidonis, A., and Goutis, C. E., *Efficient implementation of the keyed-hash message authentication code (HMAC) using the SHA-1 hash function*. in *Electronics, Circuits and Systems, 2004. ICECS 2004. Proceedings of the 2004 11th IEEE International Conference on*. 2004.
56. Siddiqui, B., *Exploring XML Encryption, Part 1.* IBM developerWorks, 2002. **3**.
57. Mukkamala, R., and Balusani, S., *Active certificates: a new paradigm in digital certificate management*. in *Parallel Processing Workshops, 2002. Proceedings. International Conference on*. 2002.
58. Simon, E., Madsen, P., and Adams, C., *XML Digital Signature.* 2001.
59. Avila-Campillo, I., Green, T. J., Gupta, A., Onizuka, M., Raven, D., and Suciu, D., *XMLTK: An XML toolkit for scalable XML stream processing.* 2002.
60. McGregor, C., Purdy, M., and Kneale, B., *Compression of XML physiological data streams to support neonatal intensive care unit Web services*. in *e-Technology, e-Commerce and e-Service, 2005. EEE '05. Proceedings. The 2005 I.E. International Conference on*. 2005.
61. Pal, S., Cseri, I., Seeliger, O., Schaller, G., Giakoumakis, L., and Zolotov, V., *Indexing XML data stored in a relational database*. 2004. VLDB Endowment.
62. Chester, T. M., Cross-platform integration with XML and SOAP. *IT Prof.* 3(5):26–34, 2001.
63. Achard, F., Vaysseix, G., and Barillot, E., XML, bioinformatics and data integration. *Bioinformatics* 17(2):115, 2001.
64. Bagnasco, A., Chirico, M., and Scapolla, A. M., *XML technologies to design didactical distributed measurement laboratories*. in *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE*. 2002.
65. Kreger, H., *Fulfilling the Web services promise.* Commun. ACM, 2003. 46(6): p. 29–ff.
66. Jia, Z., and Jen-Yao, C., *A SOAP-oriented component-based framework supporting device-independent multimedia Web services*. in *Multimedia Software Engineering, 2002. Proceedings. Fourth International Symposium on*. 2002.
67. Chiu, K., Govindaraju, M., and Bramley, R., *Investigating the limits of SOAP performance for scientific computing*. in *High Performance Distributed Computing, 2002. HPDC-11 2002. Proceedings. 11th IEEE International Symposium on*. 2002.
68. Brown, A., Fox, B., Hada, S., LaMacchia, B., and Maruyama, H., *SOAP security extensions: Digital signature.* W3C Note, 2001.
69. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S., Unraveling the Web services web: An introduction to SOAP, WSDL, and UDDI. *Internet Comput. IEEE* 6(2):86–93, 2002.
70. Kagal, L., Finin, T., Paolucci, M., Navcen, S., Sycara, K., and Denker, G., Authorization and privacy for semantic Web services. *Intell. Syst. IEEE* 19(4):50–56, 2004.
71. Ping, Z., Zhiyong, L., Tao, Q., and Xinxing, J., *Research based on XML/SOAP BACnet and internet integration technology*. in *Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on*. 2010.
72. Ostrand, T., *White-Box Testing.* Encyclopedia of Software Engineering, 2002.
73. Tonella, P. and Ricca, F., *A 2-layer model for the white-box testing of Web applications*. in *Web Site Evolution, 2004. WSE 2004. Proceedings. Sixth IEEE International Workshop on*. 2004.
74. Tonella, P., and Ricca, F., Statistical testing of Web applications. *J. Softw. Maint. Evol. Res. Pract.* 16(1–2):103–127, 2004.
75. Yu, Y.-C., and Hou, T.-W., Utilize common criteria methodology for secure ubiquitous healthcare environment. *J. Med. Syst.* 36(3):1689–1696, 2012.
76. Touati, F., and Tabish, R., U-Healthcare System: State-of-the-Art Review and Challenges. *J. Med. Syst.* 37(3):1–20, 2013.
77. Nikooghadam, M., and Zakerolhosseini, A., Secure communication of medical information using mobile agents. *J. Med. Syst.* 36(6): 3839–3850, 2012.
78. Wu, Z.-Y., Chen, L., and Wu, J.-C., A Reliable RFID Mutual Authentication Scheme for Healthcare Environments. *J. Med. Syst.* 37(2):1–9, 2013.
79. Hsu, C.-L., and Lu, C.-F., A Security and Privacy Preserving E-Prescription System Based on Smart Cards. *J. Med. Syst.* 36(6): 3637–3647, 2012.
80. Chen, H.-M., Lo, J.-W., and Yeh, C.-K., An Efficient and Secure Dynamic ID-based Authentication Scheme for Telecare Medical Information Systems. *J. Med. Syst.* 36(6):3907–3915, 2012.
81. Wu, S., Chen, K., and Zhu, Y., A Secure Lightweight RFID Binding Proof Protocol for Medication Errors and Patient Safety. *J. Med. Syst.* 36(5):2743–2749, 2012.
82. Lee, T.-F., and Liu, C.-M., A Secure Smart-Card Based Authentication and Key Agreement Scheme for Telecare Medicine Information Systems. *J. Med. Syst.* 37(3):1–8, 2013.
83. Khan, M., and Kumari, S., An authentication scheme for secure access to healthcare services. *J. Med. Syst.* 37(4):1–12, 2013.
84. Rubio, Ó.J., Alesanco, Á., and García, J., *A robust and simple security extension for the medical standard SCP-ECG. J. Biomed. Inf.*, (0).
85. Sucurovic, S., Implementing security in a distributed web-based EHCR. *Int. J. Med. Inform.* 76(5):491, 2007.
86. Blobel, B., Nordberg, R., Davis, J. M., and Pharow, P., Modelling privilege management and access control. *Int. J. Med. Inform.* 75(8): 597–623, 2006.
87. Lekkas, D., Gritzalis, S., and Katsikas, S., Quality assured trusted third parties for deploying secure internet-based healthcare applications. *Int. J. Med. Inform.* 65(2):79–96, 2002.
88. Smith, E., and Eloff, J., Security in health-care information systems—current trends. *Int. J. Med. Inform.* 54(1):39–54, 1999.

89. Moehr, J., and McDaniel, J., Adoption of security and confidentiality features in an operational community health information network: the Comox Valley experience—case example. *Int. J. Med. Inform.* 49(1): 81–87, 1998.
90. Blobel, B., Pharow, P., Spiegel, V., Engel, K., and Engelbrecht, R., Securing interoperability between chip card based medical information systems and health networks. *Int. J. Med. Inform.* 64(2–3):401–415, 2001.
91. Chen, K., Chang, Y.-C., and Wang, D.-W., Aspect-oriented design and implementation of adaptable access control for Electronic Medical Records. *Int. J. Med. Inform.* 79(3):181–203, 2010.
92. Liu, D., Wang, X., Pan, F., Xu, Y., Yang, P., and Rao, K., Web-based infectious disease reporting using XML forms. *Int. J. Med. Inform.* 77(9):630, 2008.
93. Schweiger, R., Brumhard, M., Hoelzer, S., and Dudeck, J., Implementing health care systems using XML standards. *Int. J. Med. Inform.* 74(2–4):267–277, 2005.
94. Ruotsalainen, P., A cross-platform model for secure Electronic Health Record communication. *Int. J. Med. Inform.* 73(3):291–296, 2004.
95. Gritzalis, D., and Lambrinoudakis, C., A security architecture for interconnecting health information systems. *Int. J. Med. Inform.* 73(3):305–310, 2004.
96. Rassinoux, A. M., Lovis, C., Baud, R., and Geissbuhler, A., XML as standard for communicating in a document-based electronic patient record: a 3 years experiment. *Int. J. Med. Inform.* 70(2–3):109–115, 2003.
97. Stalidis, G., Prentza, A., Vlachos, I. N., Maglavera, S., and Koutsouris, D., Medical support system for continuation of care based on XML web technology. *Int. J. Med. Inform.* 64(2):385–400, 2001.
98. Papadakis, I., Chrissikopoulos, V., and Polemi, D., Secure medical digital libraries. *Int. J. Med. Inform.* 64(2–3):417–428, 2001.
99. Norifusa, M., Internet security: Difficulties and solutions. *Int. J. Med. Inform.* 49(1):69, 1998.