International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 1 (2019), pp. 85-88
© International Research Publication House. http://www.irphouse.com

Light-weight Implementation of SSL for Secure Remote Healthcare System

Jinsoo Hwang[2], Kichang Kim[1], Changhyun Sook[1]

[1]Department of Information and Communication Engineering, Inha University, Korea.
[2]Department of Statistics, Inha University, Korea.

Abstract

Remote healthcare system is now tracking and monitoring patients and providing necessary medical treatment to them in timely manner. From the view point of data security, remote healthcare system should satisfy three requirements: confidentiality, integrity, and authenticity. It is known that the pervasive nature of remote healthcare system makes it much easier target for malicious attackers. Various researches have been performed to strengthen the security of data on transmission or in storage. Previous techniques, however, in most cases are simple re-applying of existing security techniques such as PKI (Public Key Infrastructure), which is too heavy for the embedded small medical devices attached to the patient. We suggest to simplify SSL (Secure Socket Layer) protocol to reduce its computing power requirement and use this lighter SSL as the basis of the security system for remote healthcare system.

Keywords: Remote Healthcare System, light weight healthcare security, SSL, authenticity, integrity

1. INTRODUCTION

Healthcare system is evolving beyond the boundary of hospitals or nursing facilities with the help of Internet and fast developing embedded medical devices. Remote healthcare system is now tracking and monitoring patients and providing necessary medical treatment to them in timely manner without regard to their physical location. However as the remote healthcare system is expanding, the concern on security of the data on transmission is also growing.

Numerous security breaches have been reported in relation with remote healthcare system. Health Net of Connecticut has been sued by 446,000 plan members because of the leak of patient health record and financial information. Kaiser Permanente has been fined $200,000 due to information leaking also. It is estimated that 44,000 to 98,000 patients die because of medical error in USA hospitals which was caused by corrupted medical records [1].

From the view point of data security, remote healthcare system should satisfy three requirements: confidentiality, integrity, and authenticity. Confidentiality requirement is involved when the data is transmitted from the patient or the medical device attached to the patient to Disease Management Service system. The data contains the patient's personal information and should be encrypted and protected from disclosure. Integrity requirement is involved when the data collected in the Disease Management Service is transmitted to and stored in PHR (Personal health Record) or EHR (Electronic Health Record) storage server. Of course the confidentiality requirement in this process should be met also. The data should be stored as the original content without unauthorized modification. When corruption happens in the medical data, the result could be catastrophic and may lead to deaths of patients. Finally authenticity requirement is involved when physicians access the PHR/EHR storage server. The storage server should detect unauthorized access to prevent disclosure or unauthorized modification of medical data stored in the server.

It is known that the pervasive nature of remote healthcare system makes it much easier target for malicious attackers to gain illegal access to the medical data [2]. Various researches have been performed to strengthen the security of data on transmission or in storage. The techniques suggested, however, in most cases are simple re-applying of existing security techniques such as PKI (Public Key Infrastructure). We observe that the remote health system has its own limitation in applying the existing security techniques. The main limitation is the relatively low computing power of the physical medical devices attached to the patient. If the patient is located in his or her house and the transmission distance between the device and the gateway is small, the security of the data on transmission may not be a serious problem. In this case Bluetooth transmission with moderate encryption may be enough. However if the patient is moving around in a nursing facility and the healthcare system should track the patient day or night, the transmission distance of data could be much longer and full scale secure transport protocol such as SSL (Secure Socket Layer) should be used. The problem is that small embedded medical devices often do not have enough computing power to support such powerful protocol.

In this paper, we suggest to simplify the SSL protocol to reduce its computing power requirement. Some of the load of the SSL client is shifted over to the SSL server side. Since SSL client will run in the small medical devices and SSL server runs in the central processing system, shifting the load from the client to the server is reasonable. We explain how the shifting is possible and show preliminary experimental results that show its effectiveness. The rest of the paper is organized as follows. Section 2 surveys related researches. Section 3 explains the details of our suggested techniques and provides some preliminary but promising experimental results. Section 5 gives a conclusion.

2. RELATED RESEARCHES

Secure data transfer in remote healthcare system has been researched by numerous researchers. IHE [3] proposes a framework in which healthcare enterprises are integrated to support secure, safe and reliable data transportation. Witting [4] also suggests techniques to support safe data transfer from terminal medical devices to the central server. March [5] investigates the problem of systematic storing of personal information in emergency in a secure place such as the cloud server. Real time medical data should be handled more securely and more efficiently. Woods [6] and Amer et al. [7] develop large number of medical and healthcare service rule sets to be applied when handling sensitive medical data.

Deursen et al. [8], [9] show how health data are transferred securely to the health care provider. The user generates metadata, health data, and certificates. Metadata and health data are handed over to the healthcare provider while certificates are passed over the rule engine. The health data also is passed to the aggregation engine. Then all data are collected to the reputation engine and finally to the health care provider.

The black market value of electronic protected health information (ePHI) has been rising rapidly in recent years as credit card data drops in worth [10]. Secureauth [10] advocates strong authentication to guarantee stronger ePHI protection and safer patient care. Vendatasubramaniamk [11] declares the loss of authenticity as one of the major security challenges of remote health monitor framework. The authors explain the technical challenges in securing access to the information stored in Electronic Health Record (EHR), Electronic Medical Record (EMR), and Personal Health Information (PHI). Bhattasali et al. [12] and Kavitha et al. [13] show how the pervasive nature of remote healthcare systems can make it easy for malicious attackers to gain access to these systems.

Bhattasali et al. [2] proposes a health monitor framework in which any two entities (human to human, human to machine, machine to machine) must interact only after being validated with proper authentication proof. The authors suggest bio-authentication mechanism as possible validation tool. Similar platform has been proposed in [14], [15] in which the authentication process is extended to the terminal medical devices also. The biometric authentication authenticates the patient to the device, and the device authentication process controls the access to the home hub which in turn is connected to the telehealth record server through additional authentication process. Lim et al. [16] focuses not only on secure data transportation but also on inexpensive, yet flexible and scalable, wireless platform. They report a preliminary ECG monitoring system based on this platform.

3. LEAN AND SECURE DATA TRANSPORT IN REMOTE HEALTHCARE SYSTEM

SSL(Secure Socket Layer) is the de facto secure data transport protocol in the Internet. To guarantee confidentiality, integrity, and authenticity of the medical data being transmitted in remote healthcare system, we need SSL. However SSL is notoriously heavy-duty protocol. The encryption time required for SSL protocol during key generation phase is known to be very slow because of the computation complexity of the public key algorithm.

Fig. 1 shows the basic flow of SSL protocol.

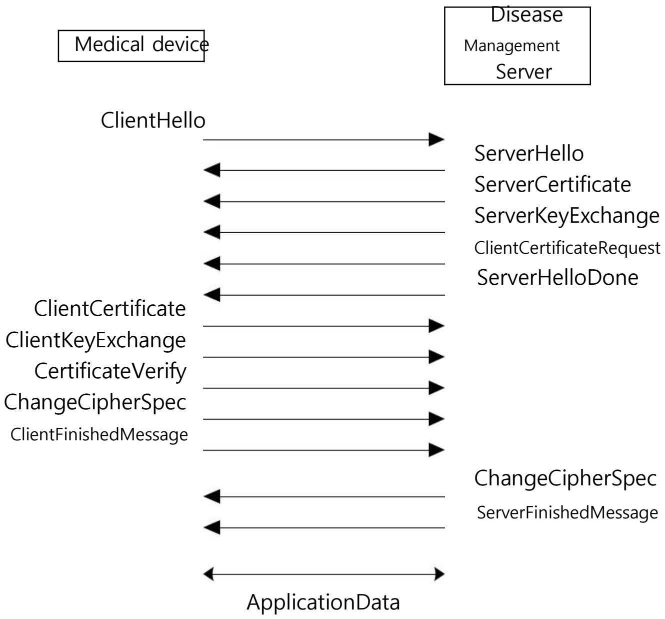

Fig. 1. Basic flow of SSL protocol

ClientHello is sent by the client (in our case embedded medical device) to the server (in our case it might be the smartphone or home gateway of the patient or the Disease Management Server) and contains various algorithm lists including encryption, compression, hashing, etc., supported by the client. ServerHello is a response to the ClientHello and is sent by the server and contains the cipher suit selected by the server. ServerCertificate contains server public key certificate and ServerKeyExchange and ClientCertificatRequest are optional. In client side, ClientCertificatae is also optional. ClientKeyExchange contains the pre-master secret generated by the client and encrypted with the server's public key. Using this pre-master secret, the server and client can generate the session key and start encrypted communication in ApplicationData phase.

We observe that ClientKeyExchange is the phase that requires high computing power due to the encryption of the pre-master secret with the server's public key. The small embedded medical devices attached to the patient, most likely, do not have the enough computing power to handle this phase. However we also observe that the generation of pre-master secret, which is the most time-consuming part in ClientKeyExchange, does not have to be done by the client. It should be perfectly OK to be performed by the server. All we need is to make sure that the client and the server receive the same pre-master secret. Below we show the modified SSL protocol step by step reflecting this idea in terms of openSSL source code.

1) Socket connection: no modification.

   The server will call ssl3_send_hello_request(SSL *s) to send SSL3_MT_HELLO_REQUEST message to the client.

2) ClientHello: no modification.

   The client will call ssl3_client_hello(SSL *s) to send SSL3_MT_CLIENT_HELLO to the server.

3) ServerHello: no modification.

   The server will call ssl3_send_server_hello(SSL *s) to send SSL3_MIT_SERVER_HELLO to the client.

4) ServerCertificate: no modification.

   The server will call ssl3_send_server_certificate(SSL *s) to send SSL3_MT_CERTIFICATE to the client.

5) Extracting server certificate: no modification.

   The client will call ssl3_get_server_certificate(SSL *s) to receive the server certificate and call X509_get_pubkey() to extract the server public key.

6) ServerKeyExchange, ClientCertificateRequest: deleted.

   The client's certificate will be sent automatically.

7) ServerHelloDone: no modification.

   The server will call ssl3_send-server_done(SSL *s)

8) ClientCertificate: no modification.

   The client will call ssl3_send_client_certificate(SSL *s) to send SSL3_MT_CERTIFICATE to the server.

9) Extracting client certificate: new addition.

   The server will call ssl3_get_client_certificate(SSL *s) to receive the client's certificate and call X509_get_pubkey() to extract the client's public key.

10) ClientKeyExchange: deleted.

    The task of generating pre-master secret will move over to the server.

11) CertificateVerify: deleted.

12) Generating 48-bit random number and pre-master secret: new addition.

    The server will generate pre-master secret, encrypt it with the client's public key and send to the client.

13) Extracting pre-master secret: new addition.

    The client will call RSA_private_decrypt() to extract pre-master secret sent from the server.

14) The rest of the protocol: no modification.

The changed part is Step 6, 9, 10, 11, 12, and 13. Step 10, 11, and 13 are for client side. The client device doesn't have to generate pre-master secret, so we delete Step 10. Since the client does not generate pre-master secret, it does not need the server's certificate which was used before to encrypt the pre-master secret, so we delete Step 11. However, the client needs to extract the pre-master secret sent by the server, so we add Step 13. Step 6, 9, and 12 are for the server side. The server doesn't have to request client certificate since the client will send its certificate automatically, so we delete Step 6. Instead, the server should generate pre-master secret and encrypt it with the client's public key (Step 12) where the client's public key is extracted in Step 9. Therefore Step 9 and 12 should be added in the server.

The changed protocol moves the burden of generating and encrypting pre-master secret from the client to the server. In general case, we have only one server and a large number of clients, and this shifting of burden will put stress on the server system. However, in remote healthcare system with small embedded medical devices, moving the load from the client to the server side is beneficial as shown in Table 1. Table 1 shows the run time of SSL protocol at client side for different SSL key size, 1024 bit and 2048 bit. The suggested modification of SSL, lean SSL, shows faster SSL connection time in both key sizes.

Table 1. Performance comparison between traditional SSL and lean SSL

	1024 bit	2048 bit
SSL	0.407	0.531
lean SSL	0.119	0.515

4. CONCLUSION

The pervasive nature of remote healthcare system makes it much easier target for malicious attackers. Various researches have been performed to strengthen the security of data on transmission or in storage. Previous techniques, however, in most cases are simple re-applying of existing security techniques, which is too heavy for the embedded small medical devices attached to the patient. This paper has suggested to simplify SSL protocol to reduce its computing power requirement and use this lighter SSL as the basis of the security system for remote healthcare system. The proposed technique shifts the burden of generating pre-master secret in SSL protocol from the client side to the server side. We have shown that this shifting reduces the load of the small medical devices considerably and enable secure data communication between embedded medical devices and DMS.

Acknowledgments

This study was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2017R1E1A1A03070865)

REFERENCES

[1] Petkovic, M.: Security challenges and technical solutions in the domain of remote patient monitoring. In: https://www.chu-toulouse.fr/IMG/pdf/Milan_Petkovic_2.pdf. (2010)

[2] Bhattasali, T., Saeed, K., Chaki, N., Chaki, R.: Bio-Authentication for Layered Remote Helath Monitor Framework. Journal of medical Informatics & Techonologies, vol. 23 (2014)

[3] Integrating the Healthcare Enterprise (IHE): IHE IT Infrastructure Technical Framework: Cross-Enterprise User Authentication (XUA) Integration Profile. http://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol1.pdf (2006)

[4] Witting, K. : Healthcare and Life Sciences: Deployment Guide – Setting up an XDS Affinity Domain using IHII Components. IBM Healthcare and Life Science, June. (2006)

[5] March, A.W.: System/method for secure storage of personal information and for broadcast of the personal information at a time of emergency. US Patent 6034605 (2000)

[6] Woods, J.:The Five Styles of Sensory Applications. Gartner Research, G00138302, Mar. (2006)

[7] Amer, M.M.M.I, Izraiq, M.I.M.A.: System with Intelligent cable-less transducers for monitoring and analyzing biosignals. European Patent Application, EP 1815784A1 (2007)

[8] Deursen, T.V., Koster, P., Petkovic, M. : Hedaquin: A reputation-based health data quality indicator. Electronic Notes in Theoretical Computer Science, Elsevier, 2008, Vol.197(2):159-167 (2008)

[9] Deursen, T.V., Koster, P., Petkovic, M.: Reliable Personal Health Records. In: Medical Informatics Europe (MIE). IOS Press, 2008, pp.484-489 (2008)

[10] Secureauth: Revolutionizing Remote Secure Access: String, Adaptive Authentication for Healthcare. https://www.secureauth.com/SecureAuth/media/Resources/WhitePapers/SA_WhitePaper_RemoteAccess_062515.pdf?ext=.pdf (2015)

[11] Venkatasubramaniamk, K., Gupta, S. K. S.: Security for Pervasive Health Monitoring Sensor Applications: In: Proceedings of International Conference on Intelligent Sensing and Information Processing (ICPSIP), 2006, pp. 197-202 (2006)

[12] Bhattasali, T., CHAKI, R., CHAKI, N.: Study of Security Issues in Pervasive Environment of Next Generation Internet of Things. n: Proceedings of CISIM 2013, Springer, LNCS, 2013, pp. 206-217 (2013)

[13] Kavitha, S.V.: A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, Elsevier, 2011, Vol. 34, No. 1, pp. 1-11 (2011)

[14] Petković, M.: Remote Patient Monitoring: Information Reliability Challenges. In: 9th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, IEEE Press, 2009, pp. 295-301 (2009)

[15] Guajardo, J., Asim, M., Petkovic, M.: Towards Reliable Remote Healthcare Applications Using Combined Fuzzy Extraction. LNCS, Dagsthul (2010)

[16] Lim, S., Oh, T.H., Choi, Y.B., Lakshman, T.: Secuirty Issues on Wireless Body Area Network for Remote Healthcare Monitoring. In: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Compuing (2010)