



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Thesis for the Degree of Master of Engineering

User's Identity Protection Scheme for Vidyanusa Game-based Learning in Junior High School



by
Kadek Restu Yani

**Interdisciplinary Program of Information Systems
The Graduate School
Pukyong National University**

February 2016

User's Identity Protection Scheme for Vidyanusa Game-based Learning in Junior High School

(중학교용 비다누사 게임 기반 학습을 위한
사용자의 신원 보호 방안)

Advisor: Prof. Kyung-Hyune Rhee

by
Kadek Restu Yani

A thesis submitted in partial fulfillment of the requirements
for the degree of

Master of Engineering

in Interdisciplinary Program of Information Systems, The Graduate School,
Pukyong National University

February 2016

User's Identity Protection Scheme for Vidyanusa Game-based
Learning in Junior High School

A thesis

by

Kadek Restu Yani

Approved by:

(Chairman) *Man-Gon Park*

(Member) *Carmadi Machbub*

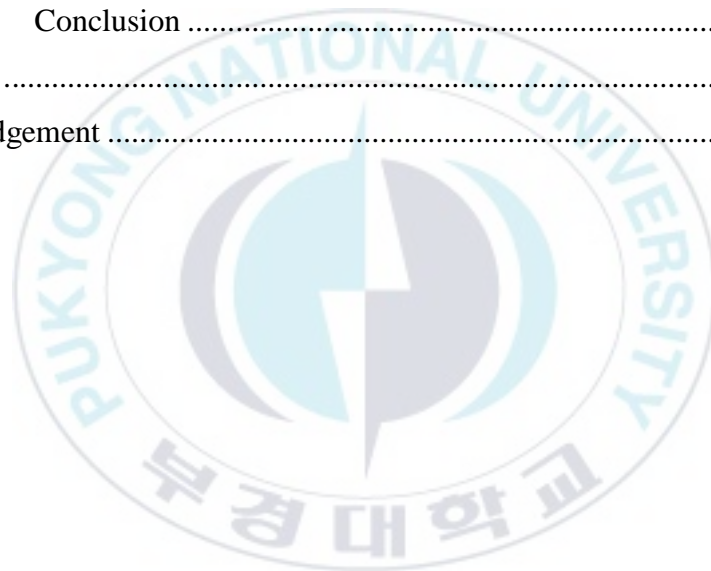
(Member) *Kyung Hyune Rhee*

February 20, 2016

Table of Contents

Table of Contents	i
List of Figures	iii
List of Tables	iv
Chapter 1. Introduction	1
1.1 Background.....	1
1.2 Problem Statement.....	2
1.3 Related Work.....	5
1.4 Thesis Objectives.....	5
1.5 Scope	6
Chapter 2. Preliminaries	7
2.1 Online Games	7
2.1.1 Game-based Learning.....	7
2.1.2 Security Issue in Online Games	8
2.2 Secure Socket Layer (SSL) Protocol	9
2.2.1 SSL handshake protocol	9
2.2.2 SSL Certificate	12
2.2.3 RSA algorithm.....	12
2.2.4 Browser security indicator: HTTPS padlock.....	14
2.3 Apache	15
2.4 OpenSSL.....	15
2.5 Wireshark.....	16
Chapter 3. System Requirements and Design.....	17
3.1 System Model	17
3.1.1 Architecture	17
3.1.2 Design Goals	18

3.1.3 The possibility of attack by malicious users.....	19
3.2 Design Model of SSL	21
3.3 Functional Security Requirements.....	25
Chapter 4. System Implementation and Analysis.....	26
4.1 Configuring an SSL Protocol	26
4.1.1 Create a Certificate	26
4.1.2 Import Certificate	30
4.1.3 Edit Apache config.....	31
4.2 Implementation of SSL Design Model.....	34
4.3 Analysis	37
Chapter 5. Conclusion	44
References	45
Acknowledgement	48



List of Figures

Figure 1. MITM attack between the browser and the server.	4
Figure 2. SSL handshake protocol.	10
Figure 3. A padlock icon appears in address bar when visiting HTTPS website.	14
Figure 4. System Model of Vidyanusa System.....	18
Figure 5. The possibility of attack by malicious users.....	20
Figure 6. Communication process between the entities.....	21
Figure 7. Generate an RSA key.	27
Figure 8. Generate a certificate sign request.....	28
Figure 9. RSA private key.	29
Figure 10. Certificate sign request.	29
Figure 11. Self-signed certificate.	29
Figure 12. Import certificate in Mozilla Firefox.	30
Figure 13. The Vidyanusa's certificate in Firefox.	30
Figure 14. Certificate viewer in Firefox.....	31
Figure 15. Add SSLRequireSSL in config file.	32
Figure 16. Enable the rewrite_module and the ssl_module.	33
Figure 17. Setup redirect HTTP to HTTPS.....	34
Figure 18. Design of Vidyanusa system to apply SSL.	35
Figure 19. The website before applying SSL connection.	35
Figure 20. The website after applying SSL connection.	36
Figure 21. The student's registration page.....	37
Figure 22. Capture SSL protocol.	38
Figure 23. The flow graph of SSL protocol.	38
Figure 24. The message of ClientHello.	39
Figure 25. The message of ServerHello.....	40
Figure 26. The message of certificate.	40
Figure 27. Application data.....	41
Figure 28. Performance HTTPS connection in web server.	42

List of Tables

Table 1. Notation used in the protocol.....	9
Table 2. Environment of Composition.....	24
Table 3. Performance Response Time of HTTPS Connection.	43



중학교용 비다누사 게임 기반 학습을 위한 사용자의 신원 보호 방안

Kadek Restu Yani

부경대학교 대학원 정보시스템협동과정

요 약

최근 웹 기반의 온라인 게임이 대화형 학습과정을 지원하기 위한 효과적이고 효율적인 수단으로 인기를 끌고 있다. 그러나 인터넷상의 안전하지 못한 채널을 통해 데이터를 전송하는 경우 개인정보의 노출이나 사용자 신원정보의 위장과 같은 보안문제를 야기하게 된다. 따라서 본 논문은 웹 기반의 온라인 게임방식의 대화형 학습시스템인 Vidyanusa 를 개발함에 있어, 학생들의 고유 코드, 사용자 식별정보와 암호를 사용하여 개인정보를 안전하게 등록할 수 있는 시스템을 개발 한다. 이를 위해, 제안 시스템은 해당 교사와 학생의 클래스 수준에 따라 사용자를 관리하고 시스템에 전송되는 사용자 식별정보의 기밀성을 위해 SSL 기반의 HTTPS 프로토콜을 이용하여 웹 서버와 클라이언트 사이의 안전한 채널을 구성하였다. 또한 Vidyanusa 시스템에 접속하는 사용자의 수에 따른 구현시스템의 성능을 분석하였으며, 실험결과로부터 SSL 을 이용한 제안시스템이 안전하고 신뢰성 있는 웹 기반의 교육시스템을 구현하는데 효과적으로 활용될 수 있음을 확인할 수 있었다.

Chapter 1. Introduction

1.1 Background

Educational gaming becomes a hot issue as an alternative tool that can be used to support the classroom learning process [1]. There are many educational online games that can be accessed for free or at a charge. Usually, online gaming needs a user identity for the registration process to ensure the legitimacy of the users who plays the game [2]. Therefore, security is needed to protect the user's identity. Additionally, elements such as score and money are included in the games, and need to be protected from malicious users [2]. Nowadays, there are many well-known threat in online gaming such as piracy, phishing attack and eavesdropping [3]. In online gaming phishing attacks are used for theft of user identity, which includes username and password, necessary to play game or manipulate gaming data [4]. In web-based online games, a phisher can produce an imitation of a website that looks similar to the legitimate one. Therefore a victim may think that is the actual website and enter his/her personal information, which is then collected by the phisher [5]. Next, eavesdropping attack is widely used by attackers in online gaming; when the attacker try to listen to all communication between the client and the server, and tries to discover the client's password and username to commit fraud [6].

In this thesis, we propose a secure online web-based educational game, Vidyanusa, which allows the users to use a unique code, besides the username, password and other information for account registration. A more detailed explanation of Vidyanusa can be found in Chapter 3. The purpose of the proposed design is to manage access of authorized users (or students) who play the game according to their school, subject

teacher and class level. Considering the threats in online gaming, the process of data sharing between the client and the server must be through a secure channel in order to protect the personal identity of student, otherwise many illegal user can play the game. An attacker may perform phishing attack using fake web pages that are highly infected with malware. The intention will bring the users to the website and extract his/her confidential details during electronic communication [6]. Moreover, the attacker may perform eavesdropping while the user starts the registration process on the system. The attacker will intercept the username and password of the users and use it to login on behalf of the victim. Our research is primarily focused on developing a secure online web-based educational game, Vidyanusa, along with securing user's identity during registration process. The security is guaranteed by using Secure Socket Layer (SSL) protocol through encryption of information exchanged via Hyper Text Transfer Protocol Secure (HTTPS) connection.

1.2 Problem Statement

Vidyanusa is an online web-based educational online game that requires a unique code as one of entity which is authenticated by the server. The unique code is given directly by the teacher to the student in the classroom. These unique codes are automatically generated by the system, when the teacher creates the class group in the dashboard site. The unique code represents the teacher identity, the class level and the class code. The purpose of adding the unique code is to authorize the students according to the subject teacher and the respective class level. Thus, the teacher has valid student data to simply manage the data mapping from the result of the game play. The results can be used for data assessment, so as to perform evaluation of students' learning ability. In addition, the teacher might discover and organize the amount of student data for each class.

In this study, we assumed that the amount of class for each grade is different, e.g., a teacher of 7th grade has three classes to teach. For this situation, the teacher first creates three classes in the system, and then the system will generate three unique codes

respective to each class. The generated unique code of the class is then shared with the legitimate students, who want to join the class. In a different case, when a student wants to register for another class then the student can register the class using a new unique code from other teacher. Meanwhile, the previous student's account will not be activated by previous teacher. Yet, the student data in the previous class is stored for backup which can be used for comparing the previous class data with the next class data. Furthermore, if the students change the school, the student can still access the system. The system administrator will update the student data such as student id, class code, and school code according to their new school, but the student's score and all data related to the game is still stored on the database server. Even though the school is changed, the students can continue to play the game as usual. The new school, which intends to use the Vidyanusa system, has to collaborate with the old school so in case if there is a student who wants to access the system, he/she can able to access the same game.

On the other hand, if the system does not generate the unique code there are some disadvantages. Firstly, a teacher can have many students in a grade (e.g. in 7th grade), although all students are not supposed to be in the same class. Moreover, it makes a complicated class on the system. When the teacher wants to create a report of the game's result for each grade, a teacher needs much effort to organize and to check the student data according to the classes. Secondly, the unique code is vulnerable to be intercepted by any attacker, since the data will be transmitted through insecure channel.

For instance, by using Man-In-The-Middle (MITM) technique, the attacker can launches an attack in the middle between a victim and a web server by replacing the legitimate URLs from server with the fake URLs as shown in Figure 1 [7]. MITM has the following steps:

- (1) The attacker launches a MITM attack (such as ARP Poisoning) to be in the middle between of victim user and web server.
- (2) When a victim requests the URL in the address bar without *https://* connection to web server, the attacker forwards a first request to the server.

- (3) The server processes the request, and returns the packets URLs.
- (4) When the attacker receives response packets from web server, such as *http://www.test.com*, it will be replaced with the fake one created by the attacker and sent to the victim.
- (5) The victim performs registration since he/she does not realize that the URLs is fake. When the victim sends any confidential information (such as username, password, and unique code), the attacker can easily eavesdrop because the packets has no encryption from HTTPS.
- (6) After the packets from the victim have arrived at the attacker's machine, the attacker then just sends it to the server.

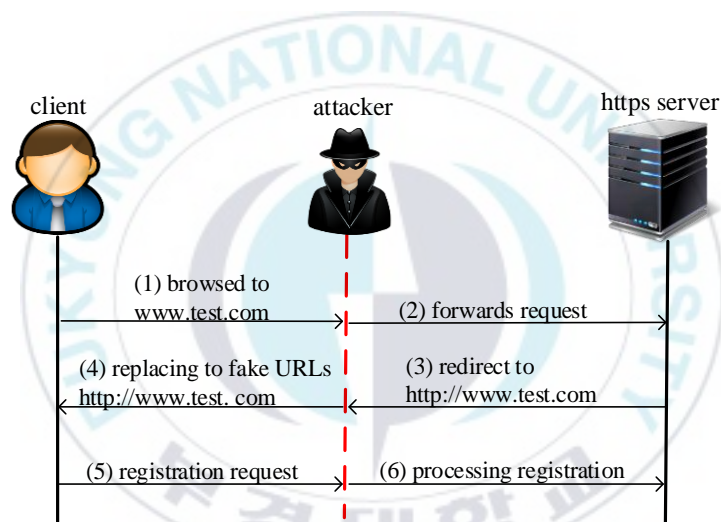


Figure 1. MITM attack between the browser and the server.

In order to provide a secure communication data exchange over the Internet, this work attempts to configure and implement an existing SSL protocol by enabling HTTPS connection in our web server.

1.3 Related Work

This section briefly explains the literature work associated with our research. Most of the existing researches use SSL protocol to provide a secure communication and data transmission over the Internet. In [8], Zi and Xu have implemented the BS1PRSA in SSL web server to improve the SSL handshake performance. A further speedup is obtained by proper use of batch technique, and shifting some decryption work to SSL clients. The theoretical value shows a substantial speedup to SSL handshake. The author in [9], enables HTTPS to protect HTTP attacks by encrypting HTTP message in the web page. The URLs of the web pages using HTTPS begin with `https://`, and by default the data is transmitted through port 443. In [10], the authors proposed a modified Email Based Identification and Authentication (EBIA) protocol for user registration. By securely accessing the Mail Servers with SSL protocol, the email account owners can specifically fetch the messages in their inbox. In [11], the researchers designed and applied an SSL protocol to protect the biometric information sent to the Hospital Information System (HIS) from mobile devices.

We intend to adopt the SSL protocol version from [8], [12] to implement in our system. To verify the user authentication, we use a unique code as one of the data that is sent to the web server during registration. The unique code is authenticated by the server to verify a legitimate user, so the data transmission will be encrypted using SSL protocol to protect the data confidentiality [12].

1.4 Thesis Objectives

This thesis has three main objectives:

1. To provide a secure data communication between client and server during registration process in Vidyanusa system (web server) over the network.
2. To guarantee the security of user's identity when the legitimate user access the system according to the unique code.

3. To protect the information including a username, a password, and a unique code by HTTPS connection from illegal users who wants to sniff the data.

1.5 Scope

By implementing the SSL protocol, the HTTPS has been designed to prevent eavesdroppers and attacker from web application services [13]. In this thesis, we focus on configuring and enabling an SSL protocol in our web server particularly the web server redirect from a normal HTTP connection to a secure HTTPS connection. Our goal is to establish a secure communication during the data transmission between a client and a server. In order to verify the legitimate user to access the system, we must keep the data confidentially.

The rest of this thesis is organized as follows: Chapter II gives a brief description of related work on educational online games and SSL connection preliminaries. Chapter III explains the system requirements and design of SSL protocol in our system. Chapter IV presents an implementation result of SSL configuration and does analysis check of the performance of HTTPS connection in the system. Finally, Chapter V summaries some concluding remarks.

Chapter 2. Preliminaries

2.1 Online Games

Online games are internet-based information technology and are considered as one type of entertainment which is widely used by people around the world [14]. The most common characteristic of online games are multiplayer gameplay which enable the users to explore and adventure in the game environment [14]. Moreover, online games is not only an entertainment, but can also function to support learning process by involving learning subjects in the games [15]. There are many kind of online games such as web-based online games which are well-known browser games. The player starts to play the game by accessing the web browser as a client [16]. The browser game is simple single player game which is played using a web browser via HTML scripting technologies such as JavaScript, PHP, and MySQL. Additionally, other development of web-based graphics technologies such as Flash or Java enhance browser games to become more complex and sophisticated [16].

2.1.1 Game-based Learning

Game-based learning employs gameplay to define learning outcomes [15]. The games are designed to balance between subject matter and gameplay; therefore the ability of the player will improve and the subject matter can easily be applied into the real world [17]. Game-based learning has been employed as a highly effective education application in the recent years. Developers generally use different models to deploy their online gaming services depending on the targeted users [18]. The purpose of using different model is to consider the content requirements analysis, game design, and to find out the security issues which are relevant to specific games [3]. Educational games are mostly developed in web gaming because of the flow of system and the content

adjusted to the learning objectives [1]. In an educational context, digital game-based learning includes activities which involve learning through problem solving. Specifically, learning arises as a result of the game's tasks, knowledge is enhanced through the game's content, and skills are developed while playing the game [19].

2.1.2 Security Issue in Online Games

The deployment of online games is growing rapidly with the advancement of the Internet. However, the growth of online games is obviously increasing along with the existence of malicious users [6], [20]. Some key threats in online games are described below.

1. Phishing attack. It is an effort to acquire sensitive information by masquerading as a trustworthy entity in an electronic communication [4]. Phishing is used to put illusion to the user who might not be able to recognize that the visited site is not real. Thus giving the hacker a chance to access personal information such as passwords, usernames, and security codes when it occurs. Web spoofing is one of the techniques used in phishing attack [21]. By forging of a website which looks similar to the victim's intended website, the attacker can collect user's password and personal information [21].
2. Eavesdropping. It is an event when users send their password and username to the server in plain text, then an eavesdropper can tap information in the middle of transmission, causing it to be compromised [3]. Usually, an eavesdropper performs on the network layer which may utilize sniffer program to tap valuable information. Packet sniffer is a program which monitor the data traveling over a network. The data is easily captured by sniffer because most of online games send the password in plain text. The attacker can intrude a system and damage it after the sniffer retrieves that data [6].

2.2 Secure Socket Layer (SSL) Protocol

The Secure Socket Layer (SSL) is a protocol which can guarantee a privacy and the authenticity of data exchanged between a client and a server in the web browser [22]. The SSL protocol uses an encrypted negotiate secret key in the SSL handshake protocol to protect communication privacy on the Internet [8]. The goal of the SSL protocol is to secure a communication between a client and a server over the network. The protocol is designed to prevent eavesdropping or message forgery [12]. SSL protocol is located between application layer and transport layer. It provides confidentiality, integrity, and authentication during data transmission [11].

2.2.1 SSL handshake protocol

An SSL protocol is a well-known handshake process which does the initial process to establish an SSL connection before a client and a server perform data exchange [8]. It aims to choose a cipher suite of two parties which includes key exchange and encryption algorithm for data exchange [8]. We adopt the standard handshaking protocol as shown in Figure 2. The SSL handshake protocol enables the server and client to authenticate each other. It also enables an encryption negotiation algorithm and using cryptographic key before the application protocol transmits or receives the first byte of data [22]. The handshake protocol assumes that, the client does not authenticate their identity to the server, only the server's identity is authenticated to the client using the server's certificate [8]. Table 1 shows the notations used in the protocol and the details of the process.

Table 1. Notation used in the protocol.

Notations	Descriptions
r_c, r_s	nonce number of a client and a server respectively
x	pre-master secret key

Notations	Descriptions
k	the shared master secret key
f	hash function
c	cipher text
e, d	encryption and decryption

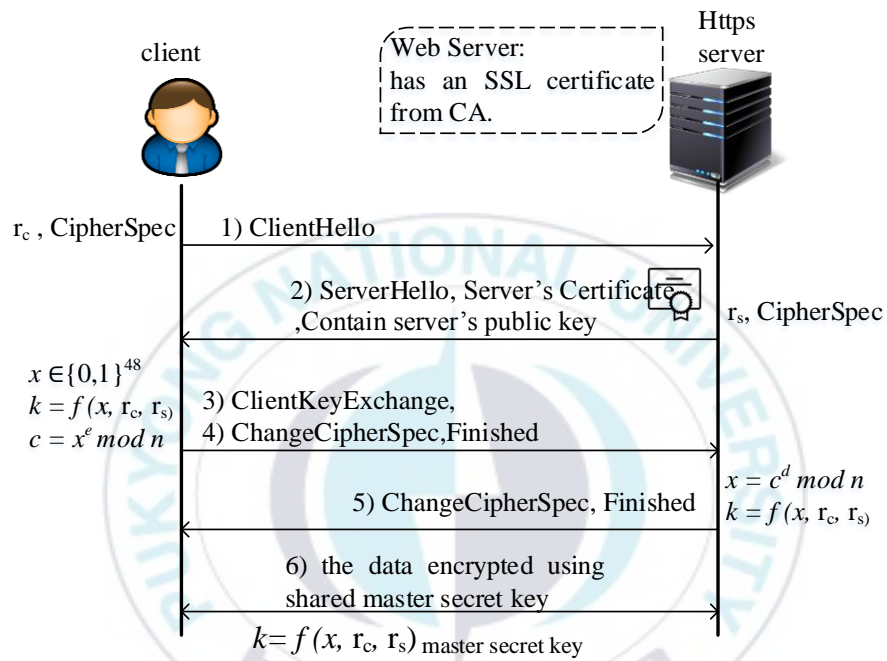


Figure 2. SSL handshake protocol.

1. ClientHello

The client sends a *ClientHello* message to the server. This indicates that the client wants to initialize SSL session with the server. The message includes an initial random nonce r_c and the cipher suites of a client's support. The random nonce is byte sequences which are chosen by a client for each connection. The client sends the list of

cipher suite which is negotiated between a server and a client. Then it will be used for a secure channel (HTTPS) connection between a client and a server.

The cipher suite is the technical protocol as a combination of authentication, encryption, a message authentication code (MAC) and key exchange algorithms. It is used when the data (plain text) is turned into cipher text, or encrypted data. For example, `SSL_RSA_WITH_RC4_128_SHA` cipher suite means that the session key will be transmitted using RSA, i.e., utilizing public key from the server certificate, it also means RC4 with a 128-bit key for data encryption is used, and then the integrity check uses the SHA-1 hash function.

2. ServerHello

The server responds with the `ServerHello` message which includes server's digital certificate, server's public key and a random nonce r_s . In addition, the server sends a cipher suites which is specified by the server choice among client candidates.

3. ClientKeyExchange

In this section, a client choose a secret random 48-byte *pre-master secret* x and computes the shared *master secret* k by inputting values of x , r_c , r_s into hash function f . Then, the client encrypts x with the server's public key and attaches the cipher text in a `ClientKeyExchange` message sent to a server. This message indicates that a client sends the negotiation of the session key to the server that should be negotiated each other for the data exchange encryption.

4. Client ChangeCipherSpec

Client also sends the `ChangeCipherSpec` message which aims to confirm the server for the subsequent of entire message within a current session which will be encrypted using key derived from a session key.

5. Server ChangeCipherSpec

After receiving the message, the server decrypts the *pre-master secret* using server's private key, and uses x value to compute the shared *master secret* k as $f(x, r_c, r_s)$. To conclude the handshake, a server responds by sending a

ChangeCipherSpec message to the client including a key hash which indicates the handshake process is finished.

6. Application data

The shared *master secret* k will be used for encrypting application data exchange during communication over the network.

2.2.2 SSL Certificate

The SSL certificate is an identity of a server which should be installed on the server to claim that the web server has trustworthy identity to perform communication. The SSL certificate is a digital certificate signed by a Certificate Authority (CA) [23]. The process of signing certificate is also known as a certificate signing request (CSR). The CSR contains information such as organization name, domain name and the public key of SSL certificate. The purpose of CSR is to create SSL certificate which will be installed on the server afterward [24]. The type of the certificate must be appropriate with the selected cipher suite's key exchange algorithm and it is generally an X.509.v3 certificate [22]. The X.509.v3 is the third version of the X.509 digital certificate standard launched in 1996 [23]. For the testing purpose and internal usage, we generate a self-signed certificate by utilizing the OpenSSL in a local web server. After finishing SSL configuration in the web server, a client can access a site securely by changing the URL from 'http://' to 'https://' [25].

2.2.3 RSA algorithm

RSA is an asymmetric cryptographic algorithm which is used to generate random number for both a public and a private key pair [26]. In RSA, the public key is used to encrypt the data and the private key is used to decrypt the data [26]. In SSL protocol, RSA is used for key agreement and authentication, the client generates a 48-byte *pre-master secret*, then encrypts it under the server's public key from server's certificate in

order to protect sensitive information during data transmission [22]. The RSA will be generated when the SSL certificate is installed on the server [24]. In this section, the steps for generating public and private key pair is described using an example which focuses on the conceptual aspects of RSA [26].

1. We assume the attribute of public key and private key as the following.

$$k_{public} = (e, n), k_{private} = (d, n)$$

2. Randomly select two prime numbers, p and q , and must be not equal. To make a strong cipher, these prime numbers should be large, and they should be in the form of arbitrary precision integers with a size of at least 1024 bits. Assume that the random values for the primes p and q are chosen:

$$p = 47 \text{ and } q = 73$$

3. Calculate the n values with $n = p * q$ (1)

$$n = 47 * 73 = 3431$$

4. Calculate the ϕ (ϕ) for these two primes. By computing with the formula

$$\phi = (p - 1). (q - 1) \quad (2)$$

$$\phi = (p - 1). (q - 1) = 3312$$

5. Next, randomly select a random integer e , with value range of $1 < e < \phi$, such that $\gcd(e, \phi)$. In this case, there is more than one possible choice and any candidate value need to be tested using the Euclidian method. Assume that we choose the following value for $e = 425$.

6. Calculate the unique integer d where, $1 < d < \phi$, the mathematical equation for this is:

$$d * e \equiv 1 \pmod{\phi} \quad (3)$$

The value of d is to be kept secret. Then the modular inverse of e is calculated to be the following: $d = 1769$.

7. Now, we have the e and d values. The e and n will make public key whereas d is private key.
8. Then, assume that the given plaintext represent by number as following: plaintext $(m) = 707$

9. For encryption, compute with the equation: $c = m^e \bmod n$ (4)

$$(cipher\ text)\ c = 707^{425} \bmod 3431 = 2142$$

10. Decryption, to recover the plaintext m from c , we compute with the following equation:

$$m = c^d \bmod n \quad (5)$$

$$(plain\ text)\ m = 2142^{1769} \bmod 3431 = 707$$

The public key is widely available to others, but the private key is concealed, only kept by the corresponding owner with the private key. The major advantage of RSA algorithm is providing authentic public key which is generally easier than distributing secret keys securely [26].

2.2.4 Browser security indicator: HTTPS padlock

HTTPS is designed to encrypt the communication and authentication between a client and a server in order to protect against HTTP attacks [7]. It is a combination of SSL protocol and HTTP to secure the web pages by encrypting HTTP messages in the transport layer protocol of open system interconnection (OSI) model [11]. By using HTTPS, confidentiality, integrity and message authentication can be provided because the HTTPS is based on an asymmetric cryptography [7]. The URLs of the web pages using HTTPS begin with `https://`, and the data are transferred over port 443 by default [9]. Figure 3 shows a padlock icon when visiting the website that provides the HTTPS connection [5].

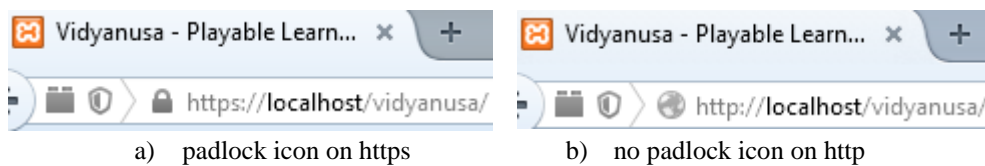


Figure 3. A padlock icon appears in address bar when visiting HTTPS website.

2.3 Apache

Apache is an open source software for web server creation, deployment and management. Apache is one of the most commonly used applications for website hosting [27]. Apache is designed to create web servers which have the ability to host one or more HTTP-based websites. Notable features include the ability to support multiple programming languages, server side scripting, an authentication mechanism and database support [28]. Apache can be extended via modules to add new useful functions such as `mod_rewrite`, which enables redirecting to control the visitor's options and shape traffic on the website. By using `mod_rewrite`, user can adjust the settings of the website traffic by editing the `httpd-xampp.conf` file which redirects all folder required to use in HTTPS connection [27].

2.4 OpenSSL

In computer networking, OpenSSL is an open-source tool of the SSL protocol for web authentication [29]. The core library, written in the C programming language, implements basic cryptographic functions and provides various utility functions such as the Base64 encoding and decoding, a symmetric encryption and decryption of files, a cryptography hashing of files, and `S_CLIENT` SSL/TLS test utility for view the information about a secure web server [29]. OpenSSL offers cryptographic functions to support SSL protocols which relies on different ciphers and algorithms to provide encryption [28]. The OpenSSL toolkit is used to generate an RSA Private Key and CSR. It can also be used to generate self-signed certificates which is used for testing purposes or internal usage [29].

2.5 Wireshark

Wireshark is an open-source network packet analyzer. Wireshark is used to analyze a communication protocol by capturing network packets data [30]. The user is able to capture either all the packets or specific packets shown in a graphical user interface (GUI) [31]. Wireshark runs on all popular computing platform such as Linux, UNIX, and Windows. The following are some of the features that provided at Wireshark [30]:

1. It can captures the live packet data from network interface.
2. It displays packets with very detail protocol information.
3. It filters packets based on many criteria.
4. It searches for packets based on many criteria.
5. It can save packet data captured.

Currently, many people use Wireshark for many purposes such as a network administrators use it for troubleshooting the network problems, network security engineers use it to examine security problems, and developers use it to debug protocol implementations [31]. Regarding the compatibility, Wireshark can read and process the files from a number of different products, including the sniffers, routers, and network utilities since it uses popular Promiscuous Capture Library (*libpcap*)-based capture format. The interface is easily compatible with other products using *libpcap*. Furthermore, Wireshark has the ability to read the variety of captured formats. Wireshark can automatically determine the type of file for reading and can decompress the GNU Zip (gzip) file [32]. Besides, the user interface of Wireshark is easy to use. The information displays very clearly showing the context of menu and a simple layout. The Wireshark GUI is useful for users who are utilizing it to analysis the packet data communication over the network [33].

Chapter 3. System Requirements and Design

3.1 System Model

3.1.1 Architecture

In this section, we describe the communication process between the entity (student or teacher) and web server of our system model as shown in Figure 4. The detailed description of system entity is as follows:

1. Web server: Vidyanusa system is an education ecosystem which consists of two main parts, the area for game play and dashboard page to manage all data involved in the system. We designed and developed an online web-based mathematical game for educating students, which focuses on 7th, 8th and 9th grade students in Junior High School. Our system has a storage capability and can be accessed online. We assumed that the web server is a trusted authority which has a certificate from certificate authority (CA).
2. The Teacher: A person who can manage the student account through the dashboard page. A teacher can
 - access the student's profile,
 - see the notification of students request,
 - check student's portfolio,
 - generate student's report,
 - analyze student's progress,
 - update student's score,
 - create a class.

Moreover, the teacher can share a unique code directly to the students. The unique code is generated automatically by the system when the teacher creates a class.

3. The students: The user who has access of Vidyanusa to play the game, view profile and portfolios. The student requires a unique code along with the username and the password during the registration process. The student needs to wait for the acceptance confirmation from the teacher before playing the game. Only an authorized student can play the game, which means that only student with a unique code can register and create an account in Vidyanusa system.

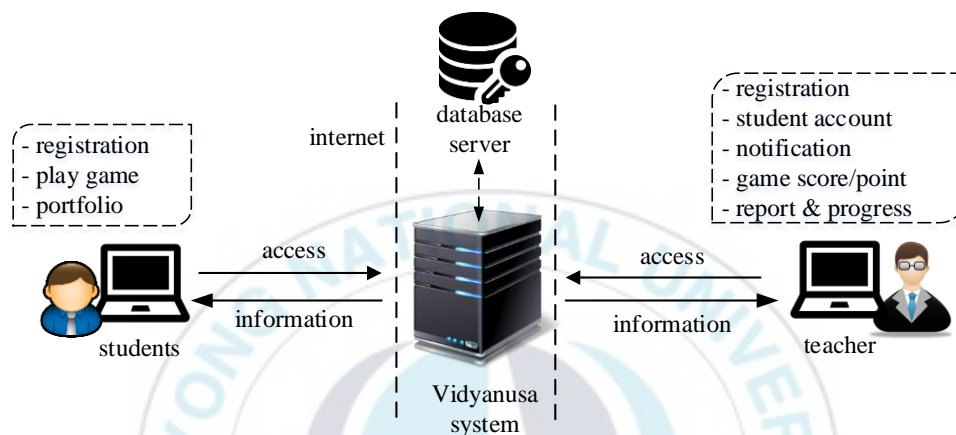


Figure 4. System Model of Vidyanusa System.

3.1.2 Design Goals

In this thesis, we consider the following purposes to propose design of Vidyanusa system:

- The game aims to increase the effectiveness and the activity of students through student-oriented learning patterns which allow student to explore the game itself. Also, to change the negative stereotype that mathematics is one of the most difficult subjects, and most students do not like it.
- The system uses a unique code during user registration to manage an authorized student that access the system according to their subject teacher. Therefore, the teacher has a valid student data and simply make a data mapping from the result of

the game play. Also, the teacher can discover and organize the amount of student data for each class easily.

- In addition, the unique code aims to manage the filtering process of the student's data in the class, so that only the students belonging to their class can join.
- The teacher can manage the student's score that will be used for measurement of students' achievement in learning mathematic through the game.

3.1.3 The possibility of attack by malicious users

In our research, the student data becomes important and should be protected during data transmission to the server through insecure network. More specific, the student uses the unique code as a main data which is authenticated by the server in order to prove whether the student is a legitimate user or not. Both students and teachers are required to create an account to access the Vidyanusa. They can login directly as student/teacher if they are already a member. However, all the users have to access the dashboard first before accessing their own page. Officially, this system is accessed by the students or the teachers in the classroom. Firstly, the teacher give a unique code directly to the students in the classroom.

Figure 5 shows the details phase of the user registration which is performed by the students, the confirm notification phase of teachers, the authentication phase of the web server and depict the possibility of attack by malicious user for ulterior purpose.

- Registration phase; after getting the unique code from the teacher, the students request the registration using the unique code and enter others information. In that case, if the students does not have a unique code, the student is not allowed to create an account.
- Authentication phase; the server will authenticate the user account by comparing a unique code from student with the one which is stored in database. After checking the data, then server send the notification to the teacher.

- Confirm notification phase; the teacher needs to confirm the student's request by accepting the notification request. After verifying the notification, the students can access the system. The students have to login first by using their username and password that already created in registration.
- The possibility of attack; attackers may launch the MITM to eavesdrop or execute the phishing attack to discover the confidential information when the legitimate students sent the data during registration to the web server. The user identity will be easy to be intercepted because the message is transmitted as a plaintext. In this case, the attacker may use the victim's account to login in Vidyanusa and can play the game while the legitimate students get an error when request for login. It will be dangerous for Vidyanusa system because the attacker can easily play the game even modify the data in the system. Moreover, the teacher will has many illegal students in their class by attacker. Even though the teacher does not realize about such condition because the attacker use a victim account.

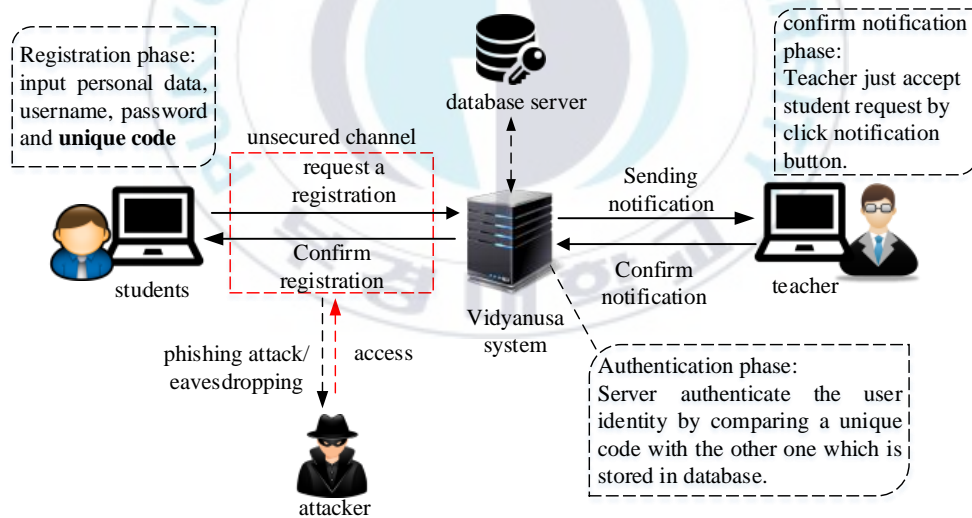


Figure 5. The possibility of attack by malicious users.

3.2 Design Model of SSL

In order to accomplish the objectives of our thesis which relies on configuring an SSL session, we adopt the standard SSL protocol in our web server. In this section, we are going to describe the handshake design model and the user registration process as shown in Figure 6. In this figure, we describe only for the students' registration process. However, either the students or the teachers are required to register before access the system. The detail of the process is as follows.

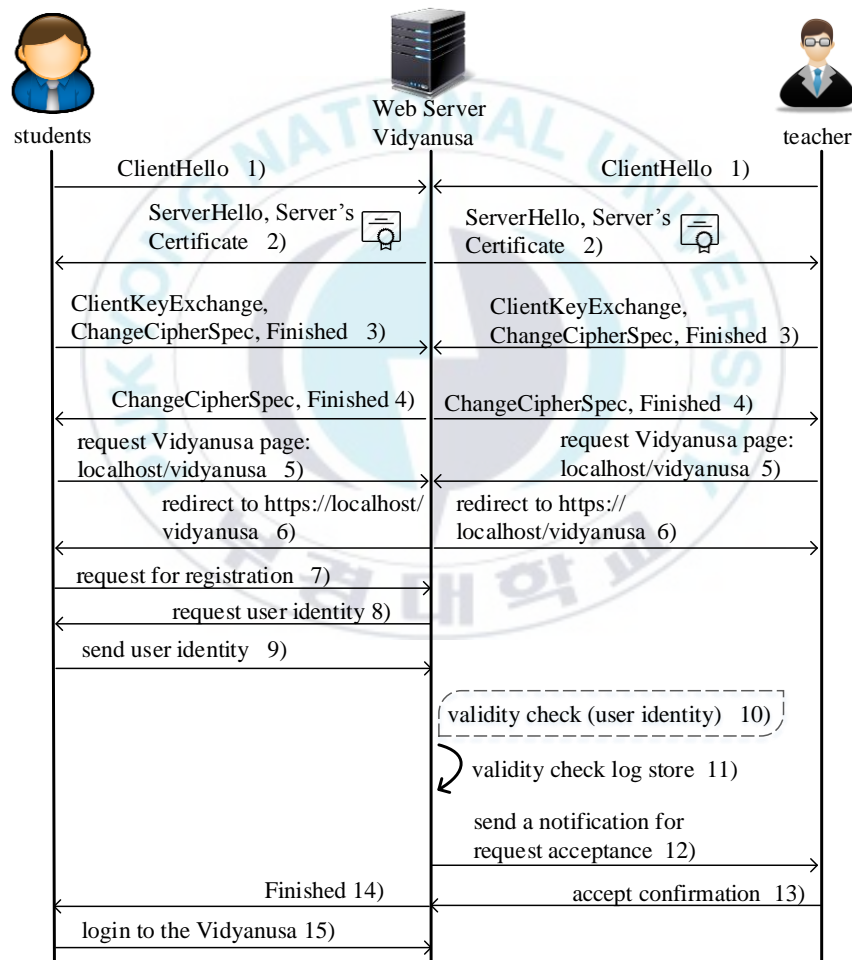


Figure 6. Communication process between the entities.

1) ClientHello

The first step is to establish communication between clients and web server. This message indicates that a client (web browser) wants to initialize an SSL session with the web server (Vidyanusa).

2) ServerHello

After the web server receives a *Hello* message from students or teachers, then the web server will respond with the *ServerHello* message that also contains of the server's certificate and the server's public key. This indicates that the server decides to establish a communication, either to the students or the teachers.

3) ClientKeyExchange, ChangeCipherSpec, Finished

In this section, the *ClientKeyExchange* message indicates that the student's web browser (client) sends the negotiation of the session key to the web server that should be negotiated each other for data exchange encryption. And also, the student's web browser will send the *ChangeCipherSpec* message that aims to notify each other that finally for the subsequent of entire message will be encrypted using the derived of session key. The same step is performed on teacher's web browser.

4) ChangeCipherSpec, Finished

When the web server responds the *ChangeCipherSpec* message of students or teachers, it means that the handshake session is finished by the server. Both entities use the session key to encrypt all data communication over the network.

5) Request Vidyanusa page: *localhost/vidyanusa*

The students start to request the Vidyanusa page. A student usually types an URL in the address bar without protocol head i.e. *localhost/vidyanusa*. The same step is performed on teacher's web browser.

6) Redirect to HTTPS connection

The web server processes the request and redirect the URLs into HTTPS connection such as *https://localhost/vidyanusa*.

7) Request for registration

When the student receives the web page information, then the student is asked to create an account for login in the system.

8) Request user identity

The web server will send the registration page and requests the student's personal data such as username, password, name, email and etc. In addition, the web server ask the students to enter the unique code and the class code will appear after the students login.

9) Send the user identity

After finishing to fulfill the data in registration form, the students then send the data to the web server.

10) Validity check (user identity)

After the server receiving the unique code, it will be authenticated in order to check the validity of the unique code is matched with the one that stored on the database server.

11) Validity check log store

The student's personal data will be stored in the database server after the server checks the validity of the data.

12) Send a notification for request acceptance

The web server send a notification message to the teacher's page. This indicates that the teacher need to check and accept the student's request for accessing the system. The notification is shows in the teacher's page.

13) Accept confirmation

The teacher accept the student's request. In this case, the teacher assumes that the students who request to join the class is the legitimate students as belonging to their class.

14) Finished

This message indicates that the students can login to the system. However, the server does not send any particular message or information in this step due to the system accessed in the same time when the classroom takes place. So, after the teacher accepts notifications, the student can login directly at that time.

15) Login to the Vidyanusa

The final step is a student login to the Vidyanusa, they can see their information page such as profile information, portfolio, the game area, and also can play that game.

Table 2 shows the analysis requirements that are needed to configure the SSL protocol in the Vidyanusa system. The web server configuration is as follows: for testing purpose, we use a localhost server using XAMPP which includes Apache that use as a web server and an OpenSSL libraries that can be utilized to create the certificate. We use MySQL 5.1.41 version for database storage. For student's browser, a Mozilla Firefox which is imported with the server's certificate is used to access Vidyanusa system. The Wireshark is used to monitoring the SSL protocol in order to capture the data packet and shows the flow graph of the data packet in Vidyanusa system.

Table 2. Environment of Composition.

		Web Server	Web Browser	S/W
Specification	CPU	Intel® Core i7- 2.93 GHz	Intel® Core i7- 2.93 GHz	-
	RAM	4.00 GB	4.00 GB	-
	Network	localhost	localhost	
OS		Windows 7 Professional	Windows 7 Professional	-
S/W		XAMPP: - Apache - OpenSSL	Mozilla Firefox	Wireshark 1.10.1
DB		MySQL 5.1.41	-	-

3.3 Functional Security Requirements

The security requirements describe the possible security services that needs to be accomplished in order to provide a secure communication data exchange over the Internet. Based on the thesis objectives, we determine the security requirements that are needed for the system are as follows:

1. User authentication: The students who wants to play the game must be authenticated by the server to ensure the user identity derived from an authorized students by comparing the data already stored on the server such as the usernames, the passwords and the unique code with the one that already sent by the students during registration. In this case, the real user identity is stored on the server to use as the authenticity when the students login to access the system again.
2. Data Confidentiality: The user identity must be protected from an unintended malicious user during the data exchange between the students and the web server over the network. By configuring an SSL protocol to secure a communication, it provides the HTTPS access to encrypt the entire messages from all client but only those authorized to have it.

Chapter 4. System Implementation and Analysis

In chapter 4, we discuss the configuration of SSL protocol in the system and the result after implementation. Moreover, we perform an analysis to capture the flow graph of data packet which is applied to an SSL connection and hence evaluate the performance of HTTPS connection.

4.1 Configuring an SSL Protocol

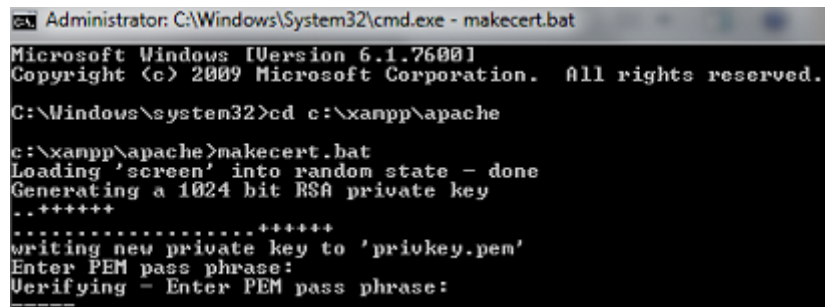
In order to enable an SSL on a web server, three steps are needed to be configured. First we need to create an SSL certificate as an identity information of our web server. Second, we need to import the certificate into the web browser that will be used to access the web server/Vidyanusa system, otherwise the web browser get a notification for untrusted certificate authority. Third, we need to edit Apache `config` for encryption that only access to the protected folder with SSL encryption. The following is a detailed process of configuring an SSL protocol.

4.1.1 Create a Certificate

We intend to create our self-signed certificate and implement in local web server. Actually, Apache provides default certificate and a key, but in this case we create a new one since the default key is available to anyone who downloads Apache. If someone knows the key, they can decrypt our packets. Generating a certificate involves three steps, which are: 1) generating an RSA private key, 2) generating a certificate sign request and, 3) generating a certificate.

1) Generating an RSA

Apache provides a batch file for creating a new certificate and a key with random encryption keys called `makecert.bat`. To execute this batch file, open a command window and do the following as show in Figure 7.



```
Administrator: C:\Windows\System32\cmd.exe - makecert.bat
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd c:\xampp\apache

c:\xampp\apache>makecert.bat
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
..+++++
.....+++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Figure 7. Generate an RSA key.

After executing the `makecert.bat` file, then the system automatically compute and generating a 1024 bit RSA private key and the system ask to enter the PEM pass phrase and at second time for a verify of a PEM pass phrase. The PEM pass phrase aims to protect private key files. We can type any word or any number to add the PEM pass phrase.

2) Generating a certificate sign request

In generating a certificate sign request, the system asks user to enter information that will be incorporated into the certificate request. As shown Figure 8, we need to input some information such as country name, province name, locality name, organization name, organizational unit name, common name and email address. Actually, we do not need to fill all information that provided in this process except, the common name. The common name must be same with the server name or the domain name of the web server. It is important that the common name match with the web address, otherwise we will get extra warnings when navigating to the secure web page.

After entering the information, the system asks user to enter the challenge password. It is an extra attribute type that specifies as a password and it sent to the certificate

request when the entity may request a certificate revocation. Figure 8 also depicts the system asking to enter the PEM pass phrase that we create in previous step. The system checks the validity of the PEM pass phrase, then the system writes an RSA key. Finally, the last command indicates that the SSL certificate is created with a private key. The `makecert.bat` script moves the server's private key and the certificate to the `conf.` file in Apache.

```

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value.
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ID
State or Province Name (full name) [Some-State]:JABAR
Locality Name (eg, city) []:BANDUNG
Organization Name (eg, company) [Internet Widgits Pty Ltd]:vidyanusa.inc
Organizational Unit Name (eg, section) []:Education Game
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:vidyanusagame@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:vidyagame
An optional company name []:vidyanusa.inc
Enter pass phrase for privkey.pem:
writing RSA key
Loading 'screen' into random state - done
Signature ok
subject=/C=ID/ST=JABAR/L=BANDUNG/O=vidyanusa.inc/OU=Education Game/CN=localhost/
emailAddress=vidyanusagame@gmail.com
Getting Private key
 1 file(s) moved.
 1 file(s) moved.

-----
Das Zertifikat wurde erstellt.
The certificate was provided.

Press any key to continue . . .

c:\xampp\apache>

```

Figure 8. Generate a certificate sign request.

3) Generating a certificate

For generating a certificate, the process is as shown in Figure 8. After the system shows the command of `Signature ok`. It indicates that our certificate is already created. By checking in the `conf.` folder, there are three files that we have which are, the RSA private key, a certificate sign request and the self-signed certificate respectively as shown in Figure 9, 10, and 11.

```

1 -----BEGIN RSA PRIVATE KEY-----
2 MIICXgIBAAKBgQCypxihSoq7xsgKtkVFkF4JqZGerZ2pMGvkT9WGU7gnzRKLIAE/
3 F+EyBnS+ZEYLF1PTfVKqVLHF/0x1JzC38M+kJcu2lw0ju+XPMVD7VI8/p0KdJcWS
4 IoyBqO3PN8P7M414jh1ThCyMFtoNUxs1yfAbxX2RBY+1M1MGGBKMq7OJdQIDAQAB
5 AoGANA7e7XkELi2BcyWjz95+mIAx77QVogx7E/9zb/4LRoXKZqBOBK2XsHUTJbug
6 cXdtu03kfb/KQbDf+QXPe3oPdv+odBrqC0TeBNr/RhVLRGP0kp3dM1mUNMBiAU9M
7 6alBycRMAvXMA4f0WcO7XlmcGDEiWIHcXbduGZtXF6fc0ECQQDkcUlvxb1f1CoI
8 XeEXFdHdodQn5+3tTul2SJ+9y1ciUurgwK23gWiQT3rOXZi85oPjTJA96fVdvYj
9 0p0Sz4HRAkEAyDQ2EPvmJy5uEG4AL6YZcFYVQ8OaaJQS1hiUagBBaiYsgjd4Yod+
10 PQwSf3kYqUwHCr3e39J07/KQ7/ohWPKyZQJBAMV2Kz1DVptj4GVVCMUqlCCk+YR
11 ZFS+gm2sUAYhkINBZjg9yXXaWlJTRLtz8yI7YTr6GKtRpbB8YOy8G+vrlFECQQCj
12 EQPIqaQ43bSAZVh5dwzZVwRTAYtQXcHJ6JNaE4LebmrqVUwU1M+Z4a9fbL8NhgNm
13 WgikXcMpfaVYeyrShDuRAkEAmdOsF0yCCPMLZE4A2klGK1TuK/FWJUD1B26Z/d+n
14 67+P74PjHt3k7LNWFVh5xYqRULDnAnAHF4eZYc50077pQ==
15 -----END RSA PRIVATE KEY-----
16

```

Figure 9. RSA private key.

```

1 -----BEGIN CERTIFICATE REQUEST-----
2 MIIBcjbCB3AIBADAUMRIwEAYDVQQDEw1sb2NhbGhvc3QwZz8wDQYJKoZIhvcNAQEB
3 BQADgY0AMIGJAoGBAME1OyFj7K0Ng2pt51+adRAj4pCdoGOVjx1BmljVnGOMW3OG
4 kHnMw9ajibh1vB6UFHxu463oJ1wLxgqx+Q8y/rPEehAjBCspKNSq+bMvZhd4p8HN
5 YMRrKffjZzv3ns1IItw46kgTgDpAl1cMRzVGPXFimu5TnWMOZ3ooyaQ0/xntAgMB
6 AAGGhzAdBgkqhkiG9w0BCQcxEBMOQXBhY2hlIEZyaWVvZHMwDQYJKoZIhvcNAQEF
7 BQADgYEAIx0oF/i847DbQDiVQ81+Uay7RzpzmdYGVgvUoVyYvY9USB2Su3WbK9vxU
8 UQyyfLgsmUQXq1VcokC9njymv4dePvdjbxjNeYIpgCGhLYO4KKVdVeJbqTfixqt6
9 ZWu6DN5CnCW7e/gjri8kifn4TQcSOTV+0kZpYYZ1PbTkWFH25nI=
10 -----END CERTIFICATE REQUEST-----
11

```

Figure 10. Certificate sign request.

```

1 -----BEGIN CERTIFICATE-----
2 MIICsTCCAhoCCQDPaxb4E256dDANBgkqhkiG9w0BAQUFADCBnDELMAkGA1UEBhMC
3 SUQXDJAMBGNVBAgMBUpBQkFSMRAWdgYDVQQHDAcCQU5EVUSHMRYwFAYDVQQKDA12
4 aWR5YW51c2EuaW5jMRcwFQYDVQQLDA5FZHVjYXRpb24gR2FtZTESMBAGA1UEAwWJ
5 bG99jYWxob3N0MSYwJAYJKoZIhvcNAQkBFhd2aWR5YW51c2FnYW11QGdtYWlsLmNv
6 bTAeFw0xNTA4MjMxMzQ0MjEwMDAwFjEwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
7 RDEOMAwGA1UECwFScFQVIXEDAOBgNVBACMB0JBTkRVTkcxRjFjAUBGNVBAoMDXZp
8 ZHlhbVZzYS5pbmMxZzAVBgNVBAsMDkVkdWNhdG1vbiBHYW11MRIwEAYDVQQDDA1s
9 b2NhbGhvc3QxJjAkBgkqhkiG9w0BCQEF3ZpZHMwDQYJKoZIhvcNAQEF3ZpZHMw
10 MIGFMA0GCSqGSIb3DQEBQUAA4GNADCBiQKBgQCypxihSoq7xsgKtkVFkF4JqZGe
11 rZ2pMGvkT9WGU7gnzRKLIAE/F+EyBnS+ZEYLF1PTfVKqVLHF/0x1JzC38M+kJcu2
12 lw0ju+XPMVD7VI8/p0KdJcWSIoyBqO3PN8P7M414jh1ThCyMFtoNUxs1yfAbxX2R
13 BY+1M1MGGBKMq7OJdQIDAQABMA0GCSqGSIb3DQEBQUAA4GBAFmz1YXMOhWlpCCE
14 3QPXW0uYk9BX9tgggtqfF8q2TdG+UEUWxrvZ1k+8W/YkID8/R01CMhiC1+J7eACy
15 291UqPUB5uGLbiwX1xDL6Kq9kcyord8iUE/bOBBeo5Au7ykpAH6YJ8hahWisgLXx
16 bKogB94eH1R8fdCkppxrq5AZPHcR
17 -----END CERTIFICATE-----

```

Figure 11. Self-signed certificate.

4.1.2 Import Certificate

Since the certificate is self-signed, and is not signed by a certificate authority (CA), the certificate must be imported into a browser that will be used to access the web server. For testing, we use a Mozilla Firefox 40.0.2 version. Figure 12 depicts the steps to import the self-signed certificate in Mozilla Firefox.

1. Open menu in Firefox → Options
2. Advanced → Certificates Menu → View Certificates Button
3. Authorities Tab → Import Button
4. Select file: c:\xampp\apache\conf\ssl.crt\server.crt, and click “Open”
5. Check “Trust this CA to identify web sites”
6. Click “OK”
7. Click “OK” in Certificate manager
8. Click “OK” In original Options window to get back into Firefox

Figure 12. Import certificate in Mozilla Firefox.

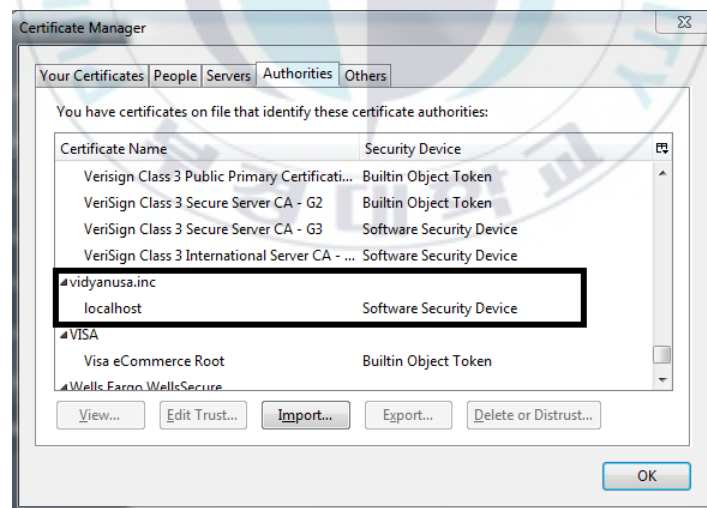


Figure 13. The Vidyanusa’s certificate in Firefox.

Figure 13 shows the self-signed certificate of Vidyanusa which is already import in the Firefox and by click the View button, we can see the full information of the certificate as shown in Figure 14.

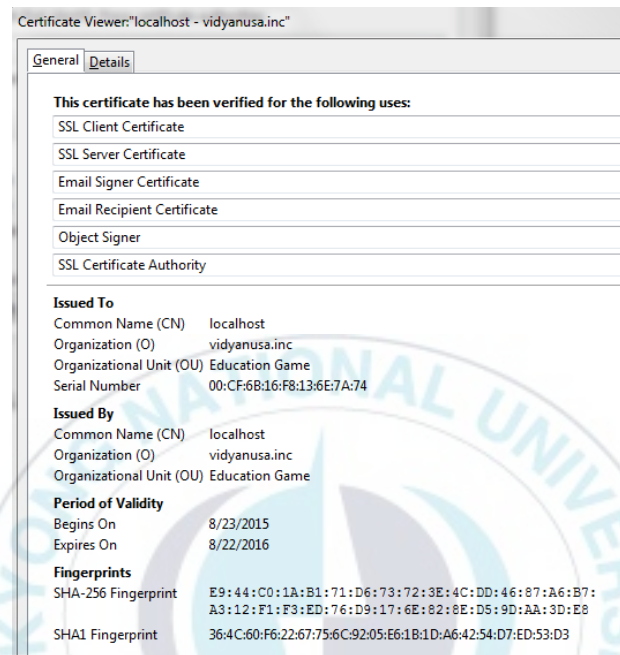


Figure 14. Certificate viewer in Firefox.

4.1.3 Edit Apache config

In Apache, we need to edit and setup `config` files for encryption to access the protected folder with SSL encryption exclusively. This is done in two steps. First, we setup the Apache `config` files for these folders to say they can only be accessed with SSL encryption. Second, we redirect any HTTP connection in these pages to the HTTPS connection.

1) Make folders accessible with SSL encryption

To make the folders accessible with the SSL encryption, we need inform Apache to encrypt the folders which is include the entire information during transmission over the network. This is accomplished by putting an *SSLRequireSSL* directive inside of each `<Directory>` listings which are required in the config file as shown in Figure 15.

```
Alias /web_folder_name "C:/xampp/foldername"
<Directory "C:/xampp/foldername">
    ...
    ...
    SSLRequireSSL
</Directory>
```

Figure 15. Add *SSLRequireSSL* in config file.

The following folders/files that we must added with the *SSLRequireSSL* which are:

- Config file: c:\xampp\apache\conf\extra\httpd-xampp.conf
 - c:\xampp\phpmyadmin
 - c:\xampp\htdocs\xampp
 - c:\xampp\webalizer
 - c:\xampp\security\htdocs
- Config file: c:\xampp\webdav
 - c:\xampp\webdav.

Next, to redirect one URL to another URL are running in the SSL connection, we also enable the modules of the *rewrite_module* and the *ssl_module* in directory c:\xampp\apache\conf by removing the hashtag (#) as shown in Figure 16.

```

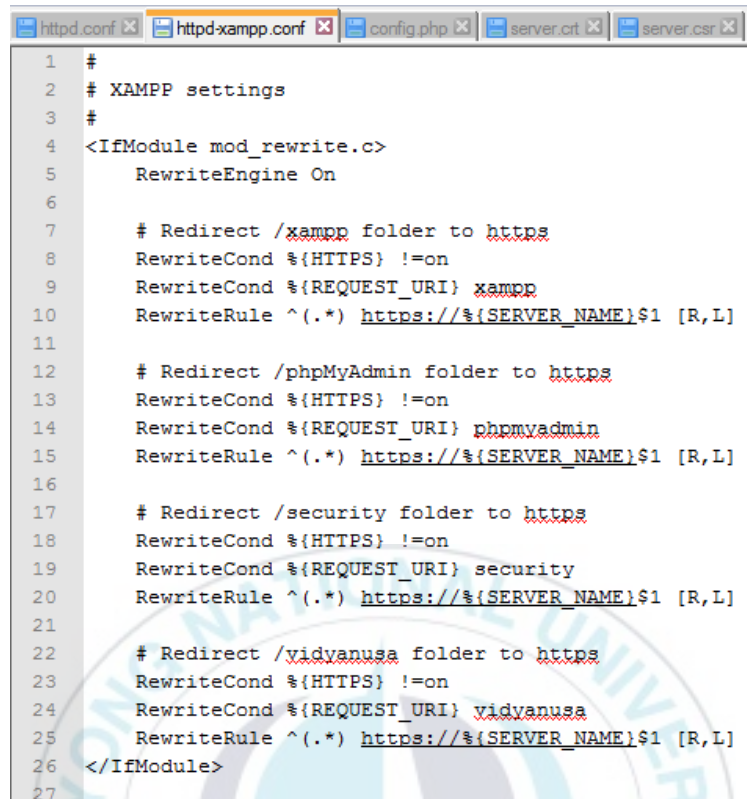
#LoadModule request_module modules/mod_request.so
#LoadModule reqtimeout_module modules/mod_reqtimeout.so
LoadModule rewrite_module modules/mod_rewrite.so
#LoadModule sed_module modules/mod_sed.so
#LoadModule session_module modules/mod_session.so
#LoadModule session_cookie_module modules/mod_session_cookie.so
#LoadModule session_crypto_module modules/mod_session_crypto.so
#LoadModule session_dbd_module modules/mod_session_dbd.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule slotmem_plain_module modules/mod_slotmem_plain.so
#LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
#LoadModule socache_dbm_module modules/mod_socache_dbm.so
#LoadModule socache_memcache_module modules/mod_socache_memcache.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
#LoadModule spelling_module modules/mod_spelling.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule status_module modules/mod_status.so
#LoadModule substitute_module modules/mod_substitute.so

```

Figure 16. Enable the `rewrite_module` and the `ssl_module`.

2) Redirect any HTTP connection to HTTPS connection

By directing the HTTP requests to the HTTPS requests for the pages that we want to protect, it aims to allow the students type the address web page without protocol head because it automatically switch to `https://`. We will use the `mod_rewrite` that already enabled in the previous step. As shown in Figure 17, we input the following source code into the top line of `httpd-xampp.conf` file from directory `c:\xampp\apache\conf\extra\httpd-xampp.conf`.



```

1  #
2  # XAMPP settings
3  #
4  <IfModule mod_rewrite.c>
5      RewriteEngine On
6
7      # Redirect /xampp folder to https
8      RewriteCond %{HTTPS} !=on
9      RewriteCond %{REQUEST_URI} xampp
10     RewriteRule ^(.*) https://%(SERVER_NAME)$1 [R,L]
11
12     # Redirect /phpMyAdmin folder to https
13     RewriteCond %{HTTPS} !=on
14     RewriteCond %{REQUEST_URI} phpmyadmin
15     RewriteRule ^(.*) https://%(SERVER_NAME)$1 [R,L]
16
17     # Redirect /security folder to https
18     RewriteCond %{HTTPS} !=on
19     RewriteCond %{REQUEST_URI} security
20     RewriteRule ^(.*) https://%(SERVER_NAME)$1 [R,L]
21
22     # Redirect /vidyanusa folder to https
23     RewriteCond %{HTTPS} !=on
24     RewriteCond %{REQUEST_URI} vidyanusa
25     RewriteRule ^(.*) https://%(SERVER_NAME)$1 [R,L]
26 </IfModule>
27

```

Figure 17. Setup redirect HTTP to HTTPS.

4.2 Implementation of SSL Design Model

We have implemented the SSL protocol in the Vidyanusa system by configuring an SSL protocol in the web server. As already mentioned above, we create an SSL self-signed certificate, import the certificates into the web browsers and make sure that the XAMPP control panel is running otherwise, the URLs cannot be access. As shown in Figure 18, the SSL connection is applied in traffic of data exchange between a client and a server using the symmetric encryption for securing the data communication.

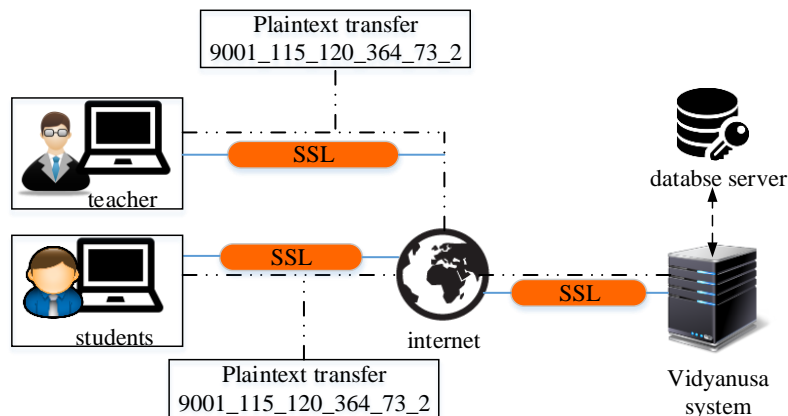


Figure 18. Design of Vidyanusa system to apply SSL.

We found different results, before and after applying SSL connection in the web server. Before applying SSL connection, the website does not provide a padlock icon and does not supply an identity information. It indicates that the website does not provides a trusted certificate and the connection is not encrypted as shown in Figure 19.

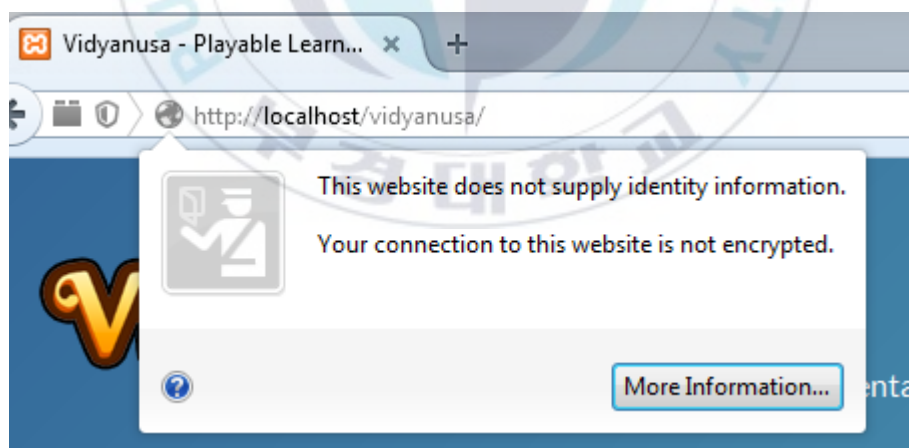


Figure 19. The website before applying SSL connection.

However, after applying SSL connection, the website provides a padlock icon and the URLs change into HTTPS in address bar. The interface on Figure 20 indicates that the certificate is already imported in the web browser. When the padlock icon is clicked by a user, the information shows the certificate issuer and the certificate is verified by vidyanusa.inc.

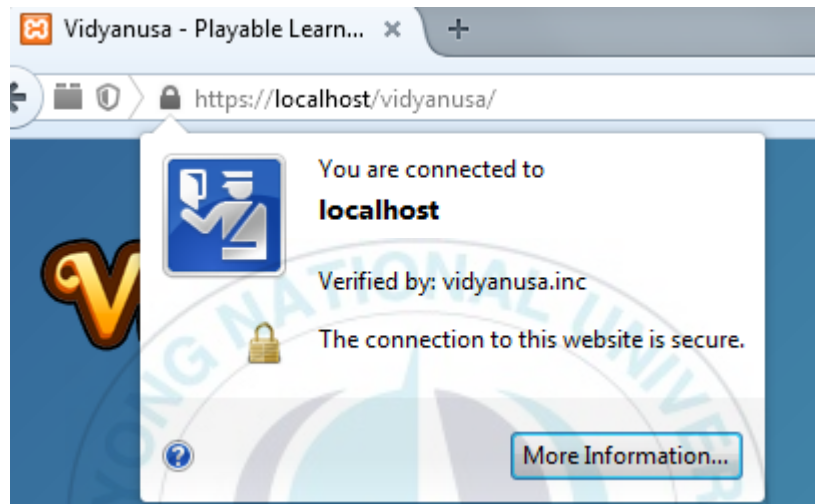


Figure 20. The website after applying SSL connection.

Figure 21 shows the list of personal information which is required when the registration process. A student enter an identity, a username, a password and a unique code. A registration success when the system verify the authenticity of a unique code is valid.



The image shows a web form titled "Student Registration Form". It contains several input fields: "Username" with the value "yani", "Name" with the value "kadek yani", "Gender" with a dropdown menu showing "Femal", "Email" with the value "olipgreensiswa@gmail.com", "Password" and "Confirm Password" both masked with dots, and "Unique Code 'Kode Masuk'" with the value "SQ1kr". There is a checkbox labeled "I declare the above data is valid" which is checked. At the bottom, there are three buttons: "Register", "Cancel", and "Already registered? Please Login". A large, faint watermark of a university logo is visible in the background.

Student Registration Form

Username
yani

Name
kadek yani

Gender
Femal

Email
olipgreensiswa@gmail.com

Password
.....

Confirm Password
.....

Unique Code "Kode Masuk"
SQ1kr

☒ I declare the above data is valid

Register Cancel Already registered? Please Login

Figure 21. The student's registration page.

4.3 Analysis

In this section, we need to make sure that the protected pages on web server are only accessed with encryption. Based on this, to monitoring the packet data which is transmitted with SSL connection, an analysis is perform using Wireshark. The analysis aims to verify the flow graph of packets data of SSL connection which is running in the system such as, information about the cipher suite, SSL version and handshake process between the client and the server.

Figure 22 shows the capture packet that is transmitted through an SSL protocol between a client and a web server. Wireshark find TLSv1.2 that used in SSL which is a new version of SSL 3.0. The handshake process start from client. The IP address of a client is 192.168.203.2 and wants to request a connection with the server which has IP

is 216.58.211.110. And also, we can see the application data is transmitted over the network. Moreover, for detail information about handshake process, Wireshark can show in the flow graph in order to easily to understand as shown in Figure 23.

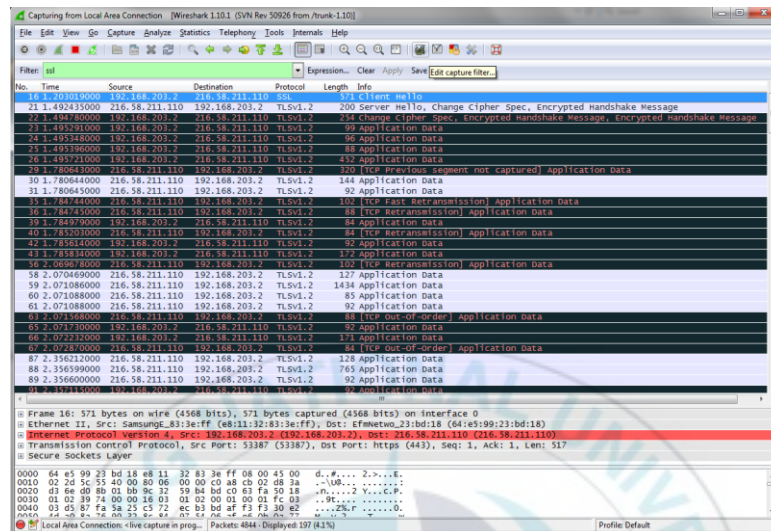


Figure 22. Capture SSL protocol.

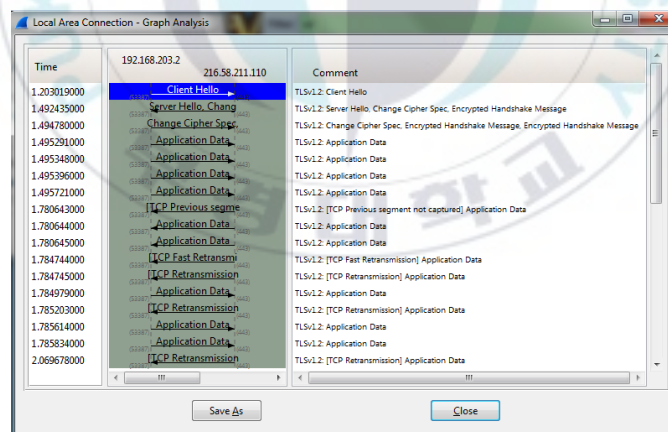


Figure 23. The flow graph of SSL protocol.

Wireshark also can verify the detail message for each packet with click the specified packet that we want to check. In this analysis, we intend to review some packet such as ClientHello, ServerHello, certificate and application data as shown in Figure 24, 25, 26, and 27 respectively. Figure 24 shows the detail information of ClientHello message. The message contains the length of cipher suite is 32, the amount of cipher suite list is 16 and a session ID of client.

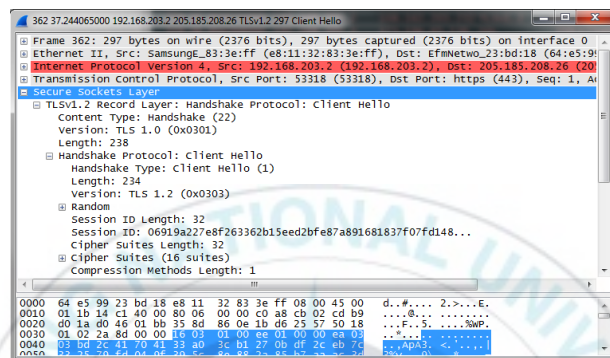


Figure 24. The message of ClientHello.

Figure 25 shows the detail information of ServerHello message. This message indicate that a server already choose one of all cipher suite which is offered from client. The cipher suite is TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 or (0xc02b) which is means that as follows.

- Key Exchange algorithm: ECDHE (Elliptic Curve Diffie-Hellman Ephemeral).
- Authentication algorithm: ECDSA (DSA with Elliptic Curve keys).
- Cipher algorithm: AES-128 (Advanced Encryption Standard with 128 bit key length).
- Message Authentication Code: SHA256 (SHA-2 algorithm with 256 bit output).

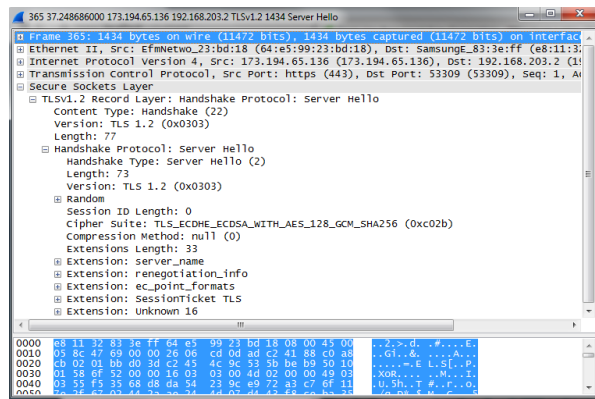


Figure 25. The message of ServerHello.

Figure 26 shows a message of certificate. This message consists of two messages, which are, the server key exchange and a certificate. The certificate is a digital identity of server's certificate and key exchange for encryption when the data transmission. Figure 27 shows the message of application data during transmission over the network which contains all information that needed between a client and a server that already encrypted.

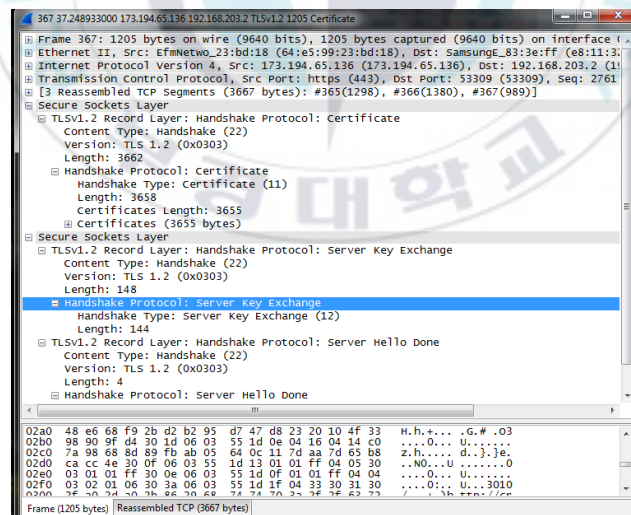


Figure 26. The message of certificate.

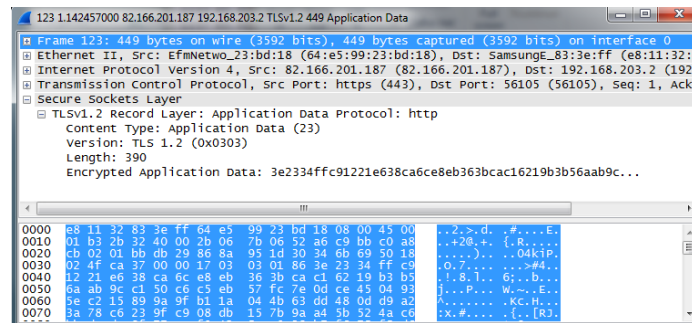


Figure 27. Application data.

By enabling SSL protocol in our system to secure communication over the network, it provides a secure connection that has three basic properties.

- The connection is private. Encryption is used after an initial handshake to define a secret key. Symmetric cryptography is used for data encryption is AES.
- The peer's identity can be authenticated using public key cryptography is ECDHE.
- The connection is reliable. When SSL certificate session is established, it provides privacy between two communicating applications. Message transport includes a message integrity check using a key of Message Authentication Code (MAC) is SHA-2 (SHA256).

Moreover, we analyze the performance of the server after applying SSL connection by sending a number of threads. A load tester tool, namely J-Meter is used in this experiment. J-Meter allows us to modelling the expected usage by simulating multiple user access on the server concurrently. We vary the students in different number such as 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. It is assumed that a student has 100 requests and has Ramp-Up period 1 second. The Rump-Up period denotes how long the next students should take to start a new session. For instance, if a server handles 100 users and 100 second of Ramp-Up, then a new session requires 1 second of waiting time before starting the new ones. Figure 28 depicts the performance of the server after applying SSL protocol.

The result shows that the performance of HTTPS as the number of users increase, the response time increases considerably. Nevertheless, the response time of HTTPS increases, it is still applicable on this web-based online game since it gains security services. In addition, HTTPS can still handle more than 50 students with 5000 requests simultaneously. The server can be accessed conveniently though 100 students perform 10000 requests at the same time. It takes only 1.318 second for the server to handle such requests. This result is supported by previous works which stated that Elliptic Curve Cryptography (ECC) provides smaller keys, making it more robust to calculate and transmit encrypted communications [34]. Although the use of basic RSA algorithm to compute the server's private key suffers from time consuming, it can guarantee the data confidentiality from malicious users [13].

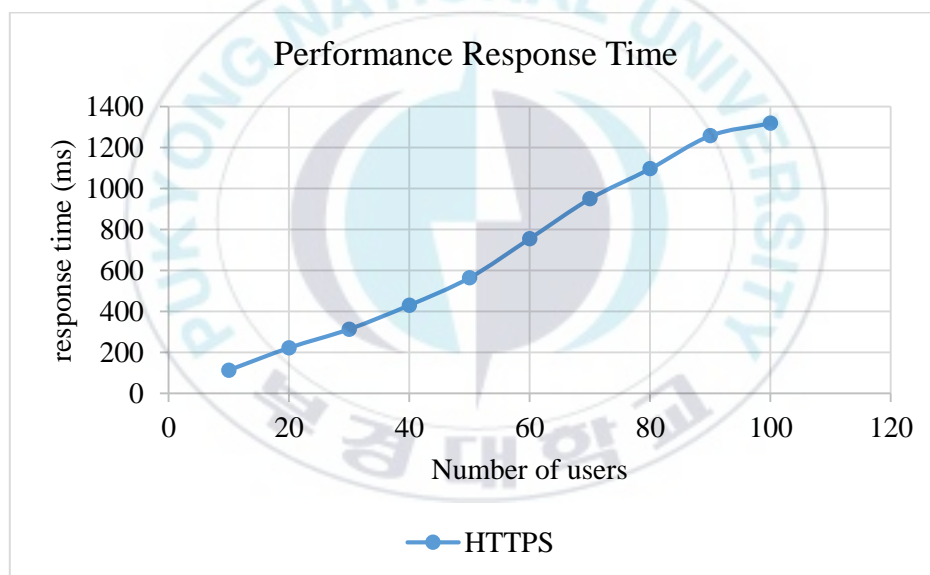


Figure 28. Performance HTTPS connection in web server.

For more detail, the performance HTTPS which is presented on the Table 3 as follows.

We use some simulation parameters which are:

1. Number of threads: 100 (number of students connect to target website).
2. Ramp-Up period: one user per 1 second.
3. Loop count: 100 (one user has 100 requests).

Table 3. Performance Response Time of HTTPS Connection.

	HTTPS									
Users	10	20	30	40	50	60	70	80	90	100
Response Time (ms)	113	222	313	430	564	755	949	1096	1257	1318
Request Samples	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
Duration Time (s)	19	26	37	50	65	83	105	119	135	144

Chapter 5. Conclusion

In this thesis we proposed the application of HTTPS connection in the web-based online game, Vidyanusa, to encrypt the entire message that sent over the internet (insecure channel). By enabling an SSL connection, it provides a secure communication that protect the student's personal information so it cannot be intercepted or modified by attackers. In contrary, the student will not be able to access the system and play the game while an attacker use the victim account to access the system at the same time.

We adopt standard SSL handshake protocol to establish SSL session for securing data transmission between the client and the web server. On the server side, we successfully configured SSL connection by creating self-signed certificate for web server internal usage. In addition, on the client side, we import the server's certificate in order to verify the identity of server for trusted authority. To verify data packet which is transmitted over SSL connection, we analyze SSL type, cipher suit, and encryption status by using Wireshark. Padlock icon will appear in the address bar of the web browser which indicate the communication is private or the communication is encrypted by SSL session. Finally, we analyze the performance of the HTTPS connection by configuring different number of user and request. HTTPS server is still reliable to handle a number of accesses simultaneously.

References

- [1] L. He, M. Fu and X. Hu, "To improve the social interaction of Web-based Collaborative Learning via online Educational Games for multi-player," in *2nd International Conference on Education Technology and Computer*, 2010.
- [2] J. Hu and F. Zambetta, "Security issues in massive online games," *Security and Communication Networks*, vol. 1, no. 1, pp. 83-92, 2008.
- [3] R. van Summeren, "Security in Online Gaming," Bachelor Thesis Information Science, Radboud University Nijmegen, 2011.
- [4] C. Wilson and D. Argles, "The fight against phishing: Technology, the end user and legislation," in *Information Society (i-Society) International Conference*, 2011.
- [5] J. Shi and S. Saleem, "Computer Security Research Reports: Phishing," University of Arizona, 2012.
- [6] J. Ki, H. J. Cheon, J. U. Kang and D. Kim, "Taxonomy of online game security," *The Electronic Library*, vol. 22, no. 1, pp. 65-73, 2004.
- [7] S. Puangpronpitag and N. Sriwiboon, "Simple and Lightweight HTTPS Enforcement to Protect Against SSL Stripping Attack," in *Computational Intelligence, Communication Systems and Networks (CICSyN), Fourth International Conference on. IEEE*, 2012.
- [8] Y. Zi and P. Xu, "The research of improving SSL handshake performance," in *Information Science and Technology (ICIST), International Conference on. IEEE*, 2013.
- [9] E. Rescorla, "HTTP over TLS–RFC 2818," in *Internet Engineering Task Force*, 2000.
- [10] J. Diaz, D. Arroyo and F. B. Rodriguez, "On securing online registration protocols: Formal verification of a new proposal," *Knowledge-Based Systems*, vol. 59, pp. 149-158, 2014.
- [11] J.-P. Lee, Y. H. Kim and J. K. Lee, "SSL Application for Managed Security between the Mobile and HIS Biometric Information Collection Client," in

*Advanced Information Networking and Applications Workshops (WAINA), 2014
28th International Conference on, 2014.*

- [12] H. Shacham and D. Boneh, Improving SSL handshake performance via batching, Berlin Heidelberg: Springer, 2001, pp. 28-43.
- [13] H. Li and G. Zhao., "Improving Secure Server Performance by EAMRSA SSL Handshakes," in *Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on. IEEE*, 2012.
- [14] C.-L. Hsu and H.-P. Lu, "Why do people play on-line games? An extended TAM with social influences and flow experience," *Information & Management*, 41(7), 853-868., vol. 41, no. 7, pp. 853-868, 2004.
- [15] K. Kiili, "Digital game-based learning: Towards an experiential gaming model," *The Internet and higher education*, vol. 8, no. 1, pp. 13-24, 2005.
- [16] A. El Rhalibi, M. Merabti, C. Carter, C. Dennett, S. Cooper, M. A. Sabri and P. Fergus, "3D Java web-based games development and deployment," in *Multimedia Computing and Systems, ICMCS'09, International Conference on IEEE*, 2009.
- [17] K. R. Yani and K. H. Rhee, "Design of a Digital Game-Based Learning Application for Junior High School," in *Proceeding of the Spring Conference of the Korea Multimedia Society*, 2015.
- [18] K. R. Yani, A. S. Prihatmanto and K. H. Rhee, "On Securing Web-based Educational Online Gaming: Preliminary Study," in *Proceeding of the Fall Conference of the Korea Information Processing Society*, 2015.
- [19] Y.-T. C. Yang, "Building virtual cities, inspiring intelligent citizens: Digital games for developing students' problem solving and learning motivation," *Computer & Education*, vol. 59, no. 2, pp. 365-377, 2012.
- [20] K. R. Yani, P. H. Rusmin and K. H. Rhee, "Applying SSL Protocol on a Web-based Educational Online Game," in *Proceeding of the Fall Conference of the Korea Information Processing Society*, 2015.
- [21] C.-Y. Huang, S.-P. Ma and K.-T. Chen, "Using one-time passwords to prevent password phishing attacks," *Network and Computer Applications*, vol. 34, no. 4, pp. 1292-1301, 2011.

- [22] A. Freier, P. Karlton and P. Kocher, "The secure sockets layer (SSL) protocol version 3.0," 2011.
- [23] K. Schmeih, *Cryptography and Public Key Infrastructure on the Internet*, John Wiley & Sons, 2006.
- [24] "What is Certificate Signing Request (CSR)?," 20 August 2015.
- [25] M. L. Das and N. Samdaria, "On the security of SSL/TLS-enabled applications," *Applied Computing and Informatics*, vol. 10, no. 1, pp. 68-81, 2014.
- [26] A. J. Menezes, P. C. v. Oorschot and S. A. Vanstone, *Handbook of applied cryptography*, CRC press, 1996.
- [27] I. Ristic, *Apache Security*, O'Reilly, 2005.
- [28] C. J. Lamprecht and A. V. Moorsel, "Adaptive SSL: Design, implementation and overhead analysis," in *Self-Adaptive and Self-Organizing Systems, 2007. SASO'07. First International Conference on*, 2007.
- [29] I. Ristic, *OpenSSL CookBook 2nd Edition*, London: Feisty Duck Digital, 2015.
- [30] U. Lamping, R. Sharpe and E. Warnicke, "Wireshark User's Guide," Interface 4, 2004.
- [31] S. Wang and S. Y. DongSheng Xu, "Analysis and application of Wireshark in TCP/IP protocol teaching," in *E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on*, 2010.
- [32] A. Orebaugh, G. Ramirez and J. Beale, *Wireshark & Ethereal Network Protocol Analyzer Toolkit*, Syngress, 2006.
- [33] C. Sanders, *Practical Packet Analysis, 2nd Edition: Using Wireshark to Solve Real-world Network Problems*, No Starch Press, 2011.
- [34] Y. Sun, X. Chen and X. Du, "An efficient elliptic curve discrete logarithm based trapdoor hash scheme without key exposure," *Journal of Computers*, vol. 8, no. 11, pp. 2851-2856, 2013.

Acknowledgement

I am grateful to the God for the good health and wellbeing that were necessary to finish my one year study in PKNu as dual degree students between PKNu and ITB. This thesis would not been complete without guidance and supporting comments from people around me, who also brace me to keep studying and make my life in Busan pleasant.

Firstly, I would like to express my sincere gratitude to my advisor Prof. Kyung-Hyune Rhee for his patience, motivation and immense knowledge. His guidance helped me in all the time of research paper and writing of this thesis. Beside my advisor, I would like to thank the rest of my thesis committee: Prof. Man-Gon Park and Prof. Carmadi Machbub, for their encouragement, insightful comments, and hard questions. My sincere thanks also goes to my advisor Professor in ITB for the guidance and the useful advice while I was studying in PKNu. My sincere thanks to Beasiswa Unggulan (BU) and SEAMOLEC for the continuous support of my Master study and related research. Also support the dual degree program, partnership between PKNu and ITB.

I thank my fellow lab mates for stimulating discussions, advices, teachings and for all fun moments. In particular, I am grateful to Dr. Park, Lewis and Bayu for enlightening me the first glance of research.

I would like to thank my family: my parents and to my brothers and sister for supporting me spiritually throughout writing this thesis and my life in general. Last but not the least, thanks to all Indonesian students in PKNu who encourage me and always give a helping hand. Also to any person or institution inside and outside whose name I could not mention here, thank you for your kind support.

