TELKOMNIKA, Vol.17, No.2, April 2019, pp.826~834
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v17i2.11754

Guillou-quisquater protocol for user authentication based on zero knowledge proof

Kevin Kusnardi*[1], Dennis Gunawan[2]
Department of Informatics, Universitas Multimedia Nusantara,
Boulevard Gading St. Serpong, Scientia Garden, Tangerang, Banten 15811, Indonesia
*Corresponding author, e-mail: kevin.kusnardi@gmail.com[1], dennis.gunawan@umn.ac.id[2]

Abstract

Authentication is the act of confirming the validity of someone's personal data. In the traditional authentication system, username and password are sent to the server for verification. However, this scheme is not secure, because the password can be sniffed. In addition, the server will keep the user's password for the authentication. This makes the system vulnerable when the database server is hacked. Zero knowledge authentication allows server to authenticate user without knowing the user's password. In this research, this scheme was implemented with Guillou-Quisquater protocol. Two login mechanisms were used: file-based certificate with key and local storage. Testing phase was carried out based on the Open Web Application Security Project (OWASP) penetration testing scheme. Furthermore, penetration testing was also performed by an expert based on Acunetix report. Three potential vulnerabilities were found and risk estimation was calculated. According to OWASP risk rating, these vulnerabilities were at the medium level.

Keywords: *cryptography, guillou-quisquater, security, user authentication, zero knowledge proof*

1. Introduction

Over the past few years, web applications continue to be a prime vector of attack for criminals [1]. Nowadays, the most common login system used in web applications sends usernames and passwords using Secure Socket Layer (SSL) [2]. The SSL approach can be exploited using SSLStrip that will intercept plain-text credentials [3]. Cryptography can be used to send confidential data through insecure channel by using encryption technique [4]. In current computer systems, cryptography provides strong economical basis for verifying integrity and keeping secret of data [5]. Cryptographic implementations can be performed using publicly accepted algorithms, such as Advanced Encryption Standard (AES), RSA for public key cryptography, and Secure Hash Algorithm (SHA-256) for hashing [6].

Based on the Internet Security Threat Report [7], at the close of 2015 there were 9 mega-breaches (over 90 million) of personal data stolen. In addition, there are also more than one million attacks on the website every day in 2015. Man-in-the-Middle is an attack where external data are injected to either hijack a data in transit or to manipulate the files and object [8]. Moreover, packet sniffing is a technique to monitor every packet which crosses the network including clear-text passwords and usernames or other sensitive materials [9]. In traditional authentication system, attacker can sniff the credentials and replay the authentication [10]. Therefore, the traditional password authentication is vulnerable to various attacks and can be easily compromised [11].

Zero Knowledge Proof (ZKP) makes a prover able to prove its identity to the verifier using a password without allowing anyone to learn anything about the password [12]. In authentication and digital signatures, it is very important to prove the user identification without revealing the user information [13]. By using ZKP, hacker who try to eavesdrop the password will be failed since the password is not sent over the insecure channel [14]. ZKP uses certificates in the authentication process. Microsoft and Google announce that SHA-1 certificates may become a risk for a website [15]. GlobalSign strongly recommends users to migrate to SHA-2 certificates as soon as possible [16].

Studies related to this research have been carried out previously. It is concluded that the main problems of Feige-Fiat Shamir protocol is the number of iterations and accreditations, which is not ideal in some implementation [17]. On the other hand, Guillou-Quisquater protocol

Received May 11, 2018; Revised November 10, 2018; Accepted November 21, 2018

takes three steps without iteration and low memory [18]. Penetration testing is a major element of all kinds of vulnerabilities for evaluating overall system [19]. It helps to secure networks and highlights the security issue [20]. It ensures the effectiveness and ineffectiveness of security measures which have been implemented by identifying and exploiting security vulnerabilities [21].

Therefore, in this study, Zero Knowledge Proof is implemented as a user authentication method using Guillou-Quisquater protocol. The login process uses two mechanisms: file-based certificate with key and local storage. Furthermore, a penetration testing is also performed based on OWASP authentication scheme and Acunetix 8.0 as a vulnerability assessment tool. The severity of the risk will also be determined based on the vulnerabilities found in the penetration testing phase.

2. Guillou-Quisquater Protocol

Guillou-Quisquater protocol improved the performance of previous zero knowledge proofs. Unlike the Fiat Shamir protocol, which uses multiple rounds and some secret values, the protocol only takes one challenge-response. The idea of this protocol is to give one difficult question, then verifier can be sure only with one correct answer [22].

This protocol requires Trusted Authority (TA) to prepare RSA cryptosystem which will be used by all parties. TA will create:

- Two primes p and q large enough that factoring their product n=p*q is infeasible.
- Another large prime b, which will be used as RSA public exponent.

The value of b and n will be published, while p and q are kept secret from all provers and verifiers [22]. Prover selects an integer as a private key u ∊ Zn * and creates a public key that satisfies (1).

$$v = (u^{-1})^b \ mod\ n \tag{1}$$

TA makes this key into certificate (using any secure signature scheme) that satisfies (2) [22].

$$\mathrm{Cert}(prover) = (\mathrm{ID}(prover), \mathrm{v}, \mathrm{sigTA}(\mathrm{ID}(prover)\ ||\ \mathrm{v})) \tag{2}$$

Here are the steps of the verification protocol.

- Prover chooses random k ∊ Zn*.
- Prover sends Cert(prover) and γ to verifier. Equation (3) shows the formula of γ.

$$\gamma = k^b \ mod\ n \tag{3}$$

- Verifier checks the certificate and rejects if verTA(ID(prover) || v, s) is false.
- Verifier sends prover random number r (0 ≤ r ≤ b−1).
- Prover computes and sends back y to verifier that satisfies (4).

$$\mathrm{y} = ku^r \ mod\ n \tag{4}$$

- Verifier accepts if γ satisfies (5).

$$\mathrm{z} \equiv v^r y^b \ (mod\ n) \tag{5}$$

The security level of this scheme is based on RSA encryption strength. The public key v is RSA encryption which is the inverse of the private key u with the key pair owned by TA. Thus, distributing v to verifier is safe because doing reverse of RSA encryption should be infeasible [22].

3. Research Method

In this research, the Guillou-Quisquater protocol is implemented into a web-based login system. Two login mechanisms are used: file-based certificate with key and local storage. The security of the login system developed will be tested by performing a penetration testing based on OWASP authentication scheme and a vulnerability assessment tool: Acunetix 8.0. The vulnerabilities found in the system are measured by an expert to estimate the risk of each vulnerability.

3.1. Design

The system flowchart is shown in Figure 1. User can choose to register or login. If the user has not registered, the user has to register first to be able to log in. Figure 2 shows the system structure. Firstly, one-time setup process will be done. The one-time setup will generate two large random prime (p and q) that factoring their product (n) is infeasible. Then the value of b (random between 0 to n) will be generated. The register process needs four data: username, password, plaintext private key, and plaintext public key. The register process will produce encrypted private key and certificate. When logging in, there are three data needed: encrypted private key, encrypted certificate, and password.

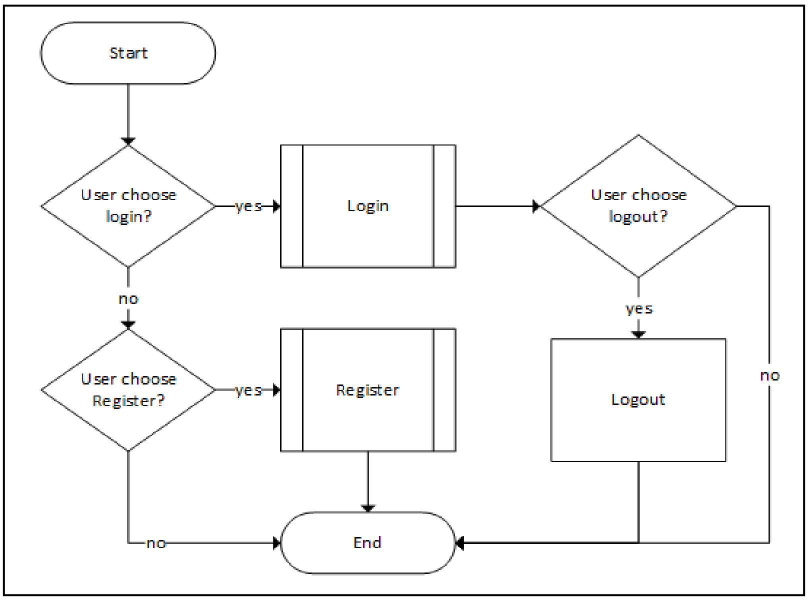

Figure 1. System flowchart

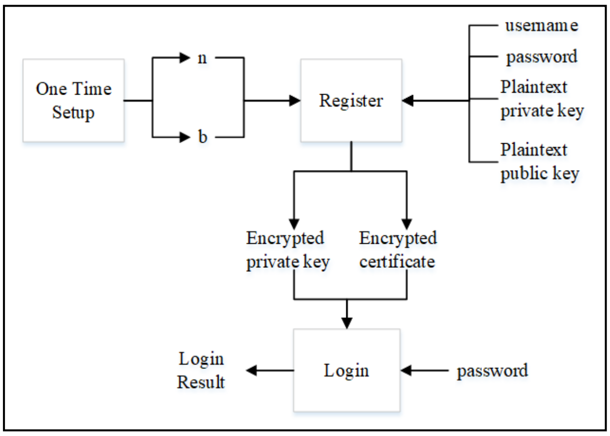

Figure 2. System structure

The register scheme is shown in Figure 3. Firstly, server will send the value of n and b from the one-time setup. After that, the client has to enter username and password. Then, the private key (random between 0 to n) and the public key that satisfies (1) will be generated on the client side. Username and public key are sent to the server and will be saved by the server. Then, the server will generate certificate with SHA-512 signature, encrypt it with AES-256, and send it to the client. The client will encrypt the private key using AES with the entered password on the client side.

Figure 4 shows the login scheme. Firstly, server will send the value of n and b. Then, client will generate k (random between 0 to n) and compute x that satisfies (3). Client will send certificate and the value of x to the server. Server will validate the certificate. If the certificate is valid, server will compute r (random between 0 to b-1). The value of r and n are sent to the client. Client will enter the private key and password to decrypt it. Then, the client will compute y that satisfies (4). The value of y is sent to the server and the server will compute z that satisfies (5) and compare z to x. If z equals to x, the client will be logged in. Otherwise, the login process will be failed. Then, the result will be sent to the client.

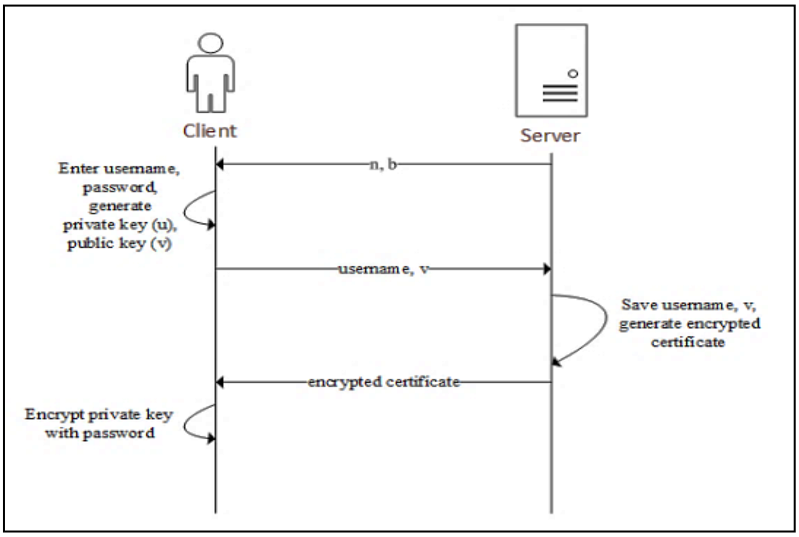
Client
Server
n, b
Enter username, password, generate private key (u), public key (v)
username, v
Save username, v, generate encrypted certificate
encrypted certificate
Encrypt private key with password

Figure 3. Register scheme

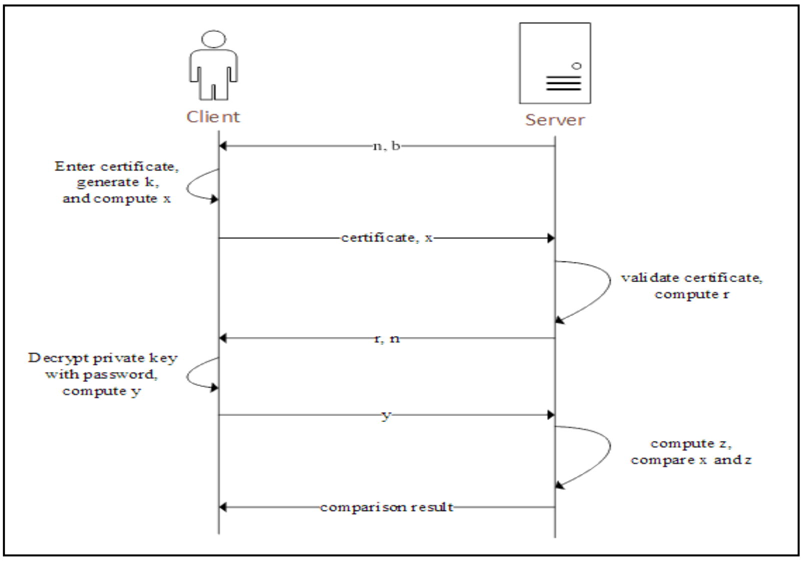
Client
Server
n, b
Enter certificate, generate k, and compute x
certificate, x
validate certificate, compute r
r, n
Decrypt private key with password, compute y
y
compute z, compare x and z
comparison result

Figure 4. Login scheme

3.2. Implementation

The Guillou-Quisquater authentication was implemented in web application. In the Register page, the user has to enter username, password, and password confirmation. After the user has already been registered, the user will have encrypted private key and certificate. This system implements two mechanisms for login: file-based certificate with key and local storage. Using the file-based login page, the user has to upload the certificate. Afterwards, the user has to upload the private key and password to decrypt the private key. On the other hand, in the login page using local storage, the user just needs to press the Next button because the certificate is taken from the local storage. As the private key is also taken from the local storage, the user just needs to enter the password to decrypt the private key.

4. Experiments and Results

Penetration testing was carried out based on OWASP scheme [23]. Besides, penetration testing was also performed by a security consultant using Acunetix 8.0.

4.1. OWASP Scheme

According to OWASP authentication scheme, the penetration testing carried out in this research consists of sniffing, username enumeration, and testing local storage.

4.1.1. Sniffing

The basic target of sniffer is to find out the password and other personal information of the user; this compromises the confidentiality [24]. Meanwhile, using Guillou-Quisquater protocol, the user will send certificate and the value of x from the result of calculation while logging in. Then, the process is followed by sending the value of y. These data are sniffed to check whether the data are sensitive. Because password is not sent to the server, these data are not sensitive. Thus, hackers cannot use these data to masquerade because the value of x and y is different on each login attempt based on the value of k which is randomized in the client side.

4.1.2. Username Enumeration

Enumeration is a process which includes active communication and direct queries to the target’s system [25]. In this section, the attempt to enumerate the registered username was carried out. The fake certificate consists of fake username (“fake user”), kevin’s public key, and signature. When logging in with the fake certificate, the login process failed, because there is no fake user in the database. Besides, the error message is not informational enough for hackers.

The other method to enumerate username is through the registration page. While registering, if the user enters an existing username, the system will generate an error message: “The username has already been taken”. Knowing existing username is not very useful for hackers, because hackers still need to know the certificate, private key, and password from that user in order to log into the system.

4.1.3. Testing Local Storage

In this section, it will be tested whether this system’s local storage can be exploited using Cross Site Scripting (XSS). Improper input validation can result in a vulnerable system against XSS attacks. This vulnerability can be exploited by hackers to retrieve local storage contents from users. An attempt to change the cert key in local storage has been made and the result is failed because the certificate is invalid. Besides, the private key was also injected in local storage and password field using JavaScript code. Beause the private key is invalid, the user is redirected to Login page. Another experiment was an attempt of XSS in password field with valid certificate and private key in local storage. The system generates an error message “Wrong key file or password” because the private key is not decrypted properly. This test shows that hackers cannot exploit the system using XSS to retrieve the local storage.

4.2. Test by Expert

The system was tested by Information Security Consultant using Acunetix 8.0. The result shows nine vulnerabilities. Six of them are authentication vulnerability, while the others are server configuration vulnerability (not discussed here). These are six vulnerabilities in the system.

a. Application Error Message

Acunetix detected an error message in the system, but the error message shown is “Whoops, looks like something went wrong”. There is no sensitive information leaked from the application error message.

b. User Credentials are Sent in Clear Text

Credentials sent to the server can be intercepted by third parties. However, on this system, confidential information such as passwords and private keys are not sent to the server. If data sent to the server is intercepted, data which can be retrieved by the third party are not confidential.

c. File Upload

If the uploaded file is not properly validated, the attacker can upload malicious files to the server to execute the code. In this system, there are two file uploads: certificate and private key. However, uploaded files sent to the server are certificates only. This certificate is validated using openssl_verify function, so it will make it harder for attacker to execute code from file upload.

d. Sensitive Page Could be Cached

Confidential information (passwords) entered in the register page can be cached using a proxy.

e. Broken Links

Some links are not accessible. However, links which are not accessible are only CSS, so the performance of this authentication system is not affected.

f. Password Type Input with Auto Complete Enabled

Acunetix found that there is a possibility of password disclosure due to the autocomplete in the active password field, but this vulnerability does not exist in the system. Therefore, this can be categorized as false positive. From the six vulnerabilities, the expert only considers three potential vulnerabilities which may become a threat to the authentication system. The three vulnerabilities are the default error message, the user credentials sent in clear text, and sensitive data exposure.

4.3. Risk Estimation

This section calculates the estimated vulnerability risk based on the penetration testing result which has been performed by the expert. The risk estimation is performed by the expert using OWASP risk estimation [23].

a. Identifying a Risk

The threat agent is everyone who opens this web page. The possible attack is sniffing and brute force. The three potential vulnerabilities are default error message, credentials sent in clear text, and sensitive data exposure.

b. Factors for Estimating Likelihood

Table 1 shows the threat agent estimation for default error message and credentials sent in clear text vulnerability, whereas Table 2 shows the vulnerability estimation for default error message and credentials sent in clear text. Table 3 shows the sensitive data exposure threat agent estimation, whereas Table 4 shows the estimation of sensitive data exposure vulnerability.

Table 1. Default Error Message and Credentials Sent in Clear Text Threat Agent Estimation

Category	Estimation
Skill level	Network and programming skills (6)
Motive	Low or no reward (1)
Opportunity	No known access (0)
Size	Anonymous internet users (9)

Table 2. Default Error Message and Credentials Sent in Clear Text Vulnerability Estimation

Category	Estimation
Ease of discovery	Automated tools available (9)
Ease of exploit	Easy (5)
Awareness	Hidden (4)
Intrusion detection	Not logged (9)

Table 3. Sensitive Data Exposure Threat Agent Estimation

Category	Estimation
Skill level	No technical skills (1)
Motive	Low or no reward (1)
Opportunity	No known access (0)
Size	Anonymous internet users (9)

Table 4. Sensitive Data Exposure Vulnerability Estimation

Category	Estimation
Ease of discovery	Easy (7)
Ease of exploit	Easy (5)
Awareness	Hidden (4)
Intrusion detection	Not logged (9)

1) Factors for Estimating Business Impact

Table 5 shows the technical impact estimation for all the three vulnerabilities.

Table 5. Technical Impact Estimation

Category	Estimation
Loss of confidentiality	Minimal non-sensitive data disclosed (2)
Loss of integrity	Minimal slightly corrupt data (1)
Loss of availability	Minimal secondary services interrupted (1)
Loss of accountability	Completely anonymous (9)

2) Determining Severity of the Risk

The overall likelihood rating is taken from the average of threat agent and vulnerability estimation, while the overall impact is taken from the average of technical impact [23]. From the calculation, the average value of likelihood from the default error message and credentials sent in clear text vulnerability is 5.375 and the likelihood of sensitive data exposure vulnerability is 4.5. Meanwhile, the average value of the technical impact of those three vulnerabilities are 3.25. The next step is to figure out whether the likelihood and impact is low, medium, or high. According to Table 6, the likelihood and impact of all vulnerabilities are in the medium level. The overall risk level is shown in Table 7. Based on the OWASP risk rating, all the vulnerabilities are in the medium level.

Table 6. Likelihood and Impact Levels [23]

Value	Level
x< 3	High
3<x<6	Medium
x>6	Low

Table 7. Overall Risk Level [23]

			Severity	
	High	Medium	High	Critical
	Medium	Low	Medium	High
Risk	Low	Note	Low	Medium
		Low	Medium	High
			Likelihood	

3) Deciding What to Fix

From the results of risk estimation, the sensitive data exposure vulnerability becomes a priority to be improved because the attacker can get the registered username. The other two vulnerabilities (the default error message and credentials sent in clear text) do not need to be fixed because no secret information can be obtained by the attacker.

5. Conclusions and Future Works

Zero knowledge proof system has been successfully implemented as an authentication process with Guillou-Quisquater protocol. Using this method, the prover can prove to verifier without revealing anything other than the fact that it knows in order to prevent the confidential information from leaking to anyone.

There are two login mechanisms implemented in this system: the file-based certificate with key and local storage. In the file-based login mechanism, the user can log into the system on different devices by uploading the certificate and private key file, while the login mechanism using local storage is more suitable if the device used for login is always the same because the certificate and private key will be taken from the browser's local storage. This makes the login mechanism using local storage become more practical than the file-based login mechanism. Based on the experiments which have been done, both mechanisms can authenticate the user correctly.

Penetration testing has been performed based on the OWASP authentication scheme. Experiments to retrieve user's private key and certificate, such as sniffing login data and XSS to retrieve local storage data, are failed. Meanwhile, the username enumeration successfully retrieves the registered username, but knowing the username is not enough to bypass the authentication system. On the other hand, the result of penetration testing by expert indicates that there are three vulnerabilities which are considered to be a threat to this system. The three vulnerabilities are the default error message, credentials sent in clear text, and sensitive data exposure. Based on the risk estimation, each vulnerability is at medium level. Based on the research which has been done, a suggestion for further research is testing the authentication system in terms of security by applying other protocols, such as Feige-Fiat-Shamir and Schnoor.

References

[1] Kaur D, Dr. Kaur P. Empirical Analysis of Web Attacks. *International Conference on Information Security & Privacy*. Nagpur. 2016; 78: 298-306.

[2] Lum J, Temasek Jun B, Polytechnic. Implementing Zero-Knowledge Authentication with Zero Knowledge (ZKA_wzk). *PyCon. Asia-Pasific*. 2010; 2: 9.

[3] Grzonkowski, Slawomir & Corcoran, Peter. A Practical Zero-Knowledge Proof Protocol for Web Applications. *Journal of Information Assurance & Security*. 2014; 8: 329-343.

[4] Acharya, K., Sajwan, M., Bhargava, S. Analysis of Cryptographic Algorithms for Network. *International Journal of Computer Applications Technology and Research*. 2014; 3(2): 130-135.

[5] IBM Knowledge Center [Internet]. ibm.com. [cited 22 July 2018]. Available from: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb300/csfb300_The_Role_of_Cryptography_in_Data_Security.htm

[6] Kenan K, Rook D, Wall K, et al. owasp.org. 2018 [cited 22 July 2018]. Cryptographic Storage Cheat Sheet. 2017. Available: https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet.

[7] Wood P (ed.), Nahorney B (ed.), Chandrasekar K, et al. Internet Security Threat Report. Symantec. Report number: 21. 2016.

[8] Olanrewaju R.F., Islam T, Khalifa OO., et al. Data in Transit Validation for Cloud Computing Using Cloud-Based Algorithm Detection of Injected Objects. *Indonesian Journal of Electrical Engineering and Computer Science*. 2018; 10(1): 348-353.

[9] Asrodia P, Patel H. Analysis of Various Packet Sniffing Tools for Network Monitoring and Analysis. *International Journal of Electrical, Electronics and Computer Engineering*. 2012; 1(1): 55-58.

[10] Dhange M, Sajjan R, Ghorpade V. A Survey on User Authentication Techniques and Attack Taxonomy in Cloud Computing. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2017; 7(2): 12-17.

[11] Jesudoss A, Subramaniam NP. A Survey on Authentication Attacks and Countermeasures in a Distributed Environment. *Indian Journal of Computer Science and Engineering*. 2014; 5(2): 71-77.

[12] Ranganathan S, Saravanan R. Password Authentication for Multicast Host Using Zero *Knowledge Proof. International Journal of Electrical and Computer Engineering*. 2015; 5(6): 1468-1471.

[13] Huqing W, Zhixin S. Research on Zero-Knowledge Proof Protocol. *International Journal of Computer Science*. 2013; 10(1): 194-200.

[14] Mohamad Z. et al. Image Based Authentication using Zero-Knowledge Protocol. 2018 4th *International Conference on Computer and Technology Applications*. 2018; 202-210.

[15] SHA-1 Hash Algorithm Migration [Internet]. symantec.com. [cited 22 July 2018]. Available from: https://www.symantec.com/theme/sha2-transition

[16] 3 Easy Steps to Migrate your Certificates from SHA-1 to SHA-256 [Internet]. globalsign.com. [cited 16 January 2018]. Available from: https://www.globalsign.com/en/blog/3-step-strategy-to-ease-the-migration-to-sha-2/

[17] Sahl AN, Samsudin A, Letchmunan S. Visual Zero-Knowledge Proof of Identity Scheme by Using Color Images. *Middle-East Journal of Scientific Research*. 2014; 21(8): 1188-1196.

[18] Guillou LC, Quisquater J. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. EUROCRYPT. 1988; 330: 123-128.

[19] Stiawan D, Idris MY, Abdullah AH, et al. Cyber-Attack Penetration Test and Vulnerability Analysis. *International Journal of Online Engineering*. 2017; 13(1): 125-132.

[20] Denis M, Zena C, Hayajneh T. Penetration testing: Concepts, attack methods, and defense strategies. *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. 2016; 1-6.

[21] Bacudio AG., Yuan X, Chu BB, et al. An Overview of Penetration Testing. *International Journal of Network Security & Its Applications*. 2011; 3(6): 19-38.

[22] Lipton B. Zero-Knowledge Proof and Authentication Protocols. 2016.

[23] OWASP Foundation. 2008 V3.0. OWASP TESTING GUIDE. OWASP Foundation. 2009.

[24] Kulshrestha A, Dubey SK. A Literature Review on Sniffing Attacks in Computer Network. *International Journal of Advanced Engineering Research and Science*. 2014; 1(2): 67-73.

[25] Alsunbul S, Le PD, Tan J. Deterring Hacking Strategies Via Targeting Scanning Properties. *International Journal of Network Security & Its Applications*. 2015; 7(4): 1-30.