Security Testing of a User-Participating Authentication Scheme Implementation on Web-Based Login System

Muhammad Akmaluddin and Yogha Restu Pramadi
Laboratory of Cryptographic Software Engineering, Sekolah Tinggi Sandi Negara/National Crypto Institute, Indonesia
* yogha.restu@stsn-nci.ac.id  * Corresponding author

Abstract. Authentication is part of the security services. From a variety of authentication methods available today, there is one scheme named user-participation authentication scheme (UPA Scheme) which not only focusing on user authentication, also provide a guarantee of human participation in the authentication process. The scheme intended for authentication in online applications such as e-learning, online polls, online ticket-order system, and more. In this study, the implementation of UPA scheme and security testing has been completed. Testing was carried out using OWASP-AT-008 with the result that the UPA scheme had guaranteed human participation, besides it also found weaknesses in the UPA scheme against man-in-the-middle (MITM) attack.

1. Introduction
One of the security services that needed is a user authentication [1]. Remote user authentication (RUA) is one model of user authentication in the online application which required to access valuable resources on the remote system [2]. In general, RUA is divided into two categories: password based authentication (PWA) and biometric-based [3]. In the PWA scheme type, there is a weakness that verification table is needed on the server [4]. To overcome this problem Hwang and Lee proposed a solution using smart cards [5]. After that, there are many RUA schemes developed using smart card. But the schemes that have been developed fails to guarantee the participation of users through the active intervention of human in the protocol [3]. Guarantee human participation can provide security on the servers from access that made by another than human. One result of the absence of the guarantee of human participation as users are bots attack. According to data obtained from Distil Networks Reports 2017 Bad Bots, there are 96 of the 100% the sites that have a login page under attack from bot [6]. Several well-known sites such as Yahoo, Popmail, and Paypal prevention that attack by using CAPTCHA [7]. Chen and Huang proposed a user authentication scheme called UPA Scheme that combining CAPTCHA and visual cryptography (VC) to ensure human participation [3]. From the explanation above, this study will be started from implementation UPA Scheme on the application of web-based login system. Further testing will be done using OWASP-AT-008 to test the CAPTCHA of UPA Scheme that have been implemented.

2. Related research
2.1. User-participating authentication scheme (UPA Scheme)
UPA Scheme combines VC and CAPTCHA to ensure the guarantee of human participation as a user. UPA schemes consists of three components: user, server, and smart cards and have 3 phases; registration, authentication, and update password.
At the registration stage, there are two steps that is:

(R.1) \( U_i \rightarrow S : ID_i, H(PW_i \oplus N) \)

(1.1) To register, user \( U_i \) selects a password \( PW_i \) and a nonce \( N \). Then count \( H(PW_i \oplus N) \)
(1.2) Then \( U_i \) send \( ID_i \) and \( H(PW_i \oplus N) \) to authentication server

(R.2) \( S \rightarrow SC : C_iH(\_\_\_) \)

(1.3) After receives \( ID_i \) and \( H(PW_i \oplus N) \), server compute \( C_1 = H(ID_i \oplus x) \oplus H(PW_i \oplus N) \) (\( x \) is the secret value stored in server) and store the value \( C_1 \) to the \( SC \) and handed it to \( U_i \).

(1.4) \( U_i \) storing the value \( N \) to the \( SC \)

The authentication phase consisting of:

(A.1) \( SC \rightarrow S : ID_i, [S_1]_{C_2 \oplus r_1}, r_1, H(S_1, r_1, C_2) \)

(1.1) \( U_i \) insert the smart card into the reader and enter a login with \( ID_i \) and \( PW_i \), the SC shall:
(1.2) Generating first visual image based shared secret \( S_1 \) and a random number \( r_1 \)
(1.3) Calculating \( H(PW_i \oplus N) \), then \( C_2 = C_1 \oplus H(PW_i \oplus N) \), and \( H(S_1, r_1, C_2) \)
(1.4) Calculating \( [S_1]_{C_2 \oplus r_1} \)
(1.5) Send parameter to the server \{ \( ID_i \), \( [S_1]_{C_2 \oplus r_1}, r_1, H(S_1, r_1, C_2) \} \)

(A.2) \( S \rightarrow SC : [S_2]_{H(ID_i \oplus x) \oplus r_2}, r_2, H(S_2, r_2, C_2) \)

After receiving a message from the user, the \( S \) operation:

(2.1) Calculating \( H(ID_i \oplus x) \oplus r_1 \), equal to \( C_2 \oplus r_1 = C_1 \oplus H(PW_i \oplus N) \oplus r_1 \), which used for calculating \( S' \) that done by decrypting \([S_1]_{C_2 \oplus r_1}\)
(2.2) Checking the hash value of \( S' \), \( r_1, C_2 \) has the same value with value received \( H(S_1, r_1, C_2) \). If the both of values same, then it will continue, otherwise it will be rejected.
(2.3) Choosing random messages \( m \) with the value of a particular digit or character.
(2.4) Generating CAPTCHA image \( img_m = CAPTCHA(m) \)
(2.5) Calculating \( S_2 = VC(S_1, img_m) \),
(2.6) Generating \( r_2 \) then calculate the symmetric key \( H(ID_i \oplus x) \oplus r_2 \) and \([S_2]_{H(ID_i \oplus x) \oplus r_2}\)
(2.7) \( S \) send \([S_2]_{H(ID_i \oplus x) \oplus r_2}, r_2, H(S_2, r_2, C_2) \) to smart cards.

(A.3) \( SC \rightarrow S : m' \)

After receiving a message from \( S \), \( SC \) performing the following operations:

(3.1) Calculate the symmetric key \( C_2 \oplus r_2 \) that as same as \( H(ID_i \oplus x) \oplus r_2 \) to get \( S' \) by decrypting \([S_2]_{H(ID_i \oplus x) \oplus r_2}\), which \( C_2 \) has been calculated in step A.1.
(3.2) Verify the hash value of \( S' \), \( r_2, C_2 \) are have the same value with the hash value received \( H(S_2, r_2, C_2) \). If the value are not same, \( SC \) will be cut off the communication, if the value are same then \( S \) is authenticated.
(3.3) \( U_i \) recognizing the visual value \( m' \) obtained by stacking (boolean) \( S_1 \) above \( S_2 \). For the record, the process of stacking \( S_1 \) and \( S_2 \) using boolean lightweight computing process will be completed by the smart card. \( U_i \) then sends \( m' \) to the \( S \). \( S \) verify, if the value \( m' \) is equal to the \( m \) then \( U_i \) authenticated with \( S \).

The last phase is update password. If user \( U_i \) want to change their password \( PW_i \) to the new password \( PW'_i \), he must perform the following operations that are not associated with the server. First enable the \( SC \) then calculate \( H(PW_i \oplus N) \) and \( H(PW'_i \oplus N) \). Then \( SC \) calculate \( C'_1 = C_1 \oplus H(PW_i \oplus N) \oplus H(PW'_i \oplus N) \) and replace \( C_1 \) with \( C'_1 \).
2.2. CAPTCHA

*Completely Automated Public Turing Test To Tell Computers and Humans Apart* (CAPTCHA) is an automated test that can differentiate between humans and computers [7]. The process is usually performed by using a server that will send a challenge to users. The challenge is a combination that generated server. Then the server will check if the response from the user is right or wrong. Because it is assumed that the other computing devices will not be able to answer the challenge CAPTCHA, then anyone who successfully answer is human. CAPTCHA testing can be done using the OWASP-AT-008 [8].

2.3. Visual Cryptography

*Visual Cryptography* (VC) is one type of cryptographic algorithms introduced by Naor and Shamir for the data in the form of image [9]. VC scheme was drawn up without requiring a complicated computation for the decryption process. Decryption process can be performed by using the human visual[9]. VC works by dividing an image data into sections then the data is returned by stacking a number of sections.

3. Implementation

The implementation process in this study guided by Software Development Life Cycle. The following hardware environments that used are PC with operating system Windows 10, smartcard reader ACR 122 U, and smart card JCOP (Java Card Open Platform) 31 2.4.1 R3. The smart card is java card type which communicating with the application using APDU (Application Data Protocol Unit). In addition in the process of implementation, AES-256 will be used as the encryption algorithm and SHA-256 will be used as a hash function. The software used in the process of implementation in this study are: JCIDE (Java Card Integration Development Environment), PyApduTool, Firefox ESR (Extended Support Release), XAMPP version 6.5.34, and WebCard plugin.

4. Analysis

The research continue to the testing phase or evaluate the applications and data that has been obtained. Tests were conducted using the method of testing for CAPTCHA (OWASP-AT-008). OWASP-AT-008 will be able to provide the results of an analysis of success in providing assurance CAPTCHA for human participation. Test started with formulating threat models and methods/techniques also the related testing tool that is used. The test categorized into two types, grey-box testing and black-box testing.

4.1. Grey-box Testing

4.1.1. Security Identification of CAPTCHA Algorithm

This test is used to identify the use of the version and type of CAPTCHA that implemented. To perform this test, assumed the attackers can get the image of CAPTCHA. Next step, the attacker try to get the value of decoded CAPTCHA or breaking the CAPTCHA image. In this tests, have been tested 5 pieces CAPTCHA image sample. From 5 samples that have been tested, there weren’t successfully solved by OCR test that performed by GSA Captcha Breaker shown at (Figure 1). It can be concluded that the CAPTCHA algorithm is safe and can prevent the activity carried out by a program or bot.
4.1.2. Security Identification of the Support Algorithm

This test used to identify the hash algorithm and the encryption used to transmit parameters associated with the CAPTCHA. In implementation process, relating to the implementation of the scheme used AES-256 algorithm for encryption and SHA-256 for hash because there are still safe to use [10][11]. With the existence of these conditions reduce the presence of security loopholes and CAPTCHA scheme that is used.

4.2. Black-box Testing

4.2.1. Security Identification for Parameters that Used to Decoding CAPTCHA

To support the testing process, Burp Suite Community Edition configured as a proxy. In this part of tests conducted security aims to identify all the parameters sended from the user to the server. These parameters will be used attacker to download and obtain decoded CAPTCHA’s ID value if it is not secured. In this test the threat model can be seen in (Figure 2). The attacker is assumed to be able to intercept the package and received response from the server after a replay the intercepted messages of user. If the parameters were not secured either with encryption or another method, will give attacker opportunity to decode CAPTCHA that transmitted at the time.

![Figure 2. Threat Model II](image)

From (Figure 3) below, the parameters passed from the client to the server are; id is the user’s id, imgBase64 is preshare image that has been encrypted with the AES algorithm, r1 is random number and shashedData is a hash value of some parameters using SHA 256 algorithm. When the four parameters sent to the server will get a response in the form of images Authentication Share encrypted using the AES algorithm, a random value from the server, and hashedData. Based on the analysis of Chen and Huang [3], attackers who intercept packets will not be able
to decode CAPTCHA because he have to decrypt preshare and Authentication Share that the key for decrypting the value is a combination of value C1 from the smart card with the user's password. Therefore we can conclude all the parameters are safe from attackers attempt to decode CAPTCHA.

![Table of POST request to /TA_FINALLogIn.php](image)

**Figure 3. Results Intercept Burp Suite**

4.2.2. Security Identification of Replay Attack and Parameter of the Decoded CAPTCHA

This test is done by doing replay attack using Burp Suite Community Edition too. The technique is sending a CAPTCHA value that has been decoded to the server along with the ID of CAPTCHA or with the value of the session ID. In experiments assumed that the attacker is able to capture packets from the client to the server. Then the attacker resending (fake replay) a CAPTCHA value that has been decoded. If the server reply with the correct response, the server vulnerable to replay attacks. Threat model of this attack against to the scheme can be found at (Figure 4). In these conditions, if the user has successfully logged on to the server the attacker can’t perform a replay attack. This condition caused that the CAPTCHA’s decrypted value only can be used one time. Consequently, replay packets from the attacker will get response "wrong captcha string" from servers such as the (Figure 5). With these conditions, it can be concluded that in this test successfully passed.

![Threat Model III](image)

**Figure 4. Threat Model III**

But, these security parameters of decoded CAPTCHA is known that are sent in an open state that shown by red marked box in (Figure 5). These conditions causing the founded any weaknesses of the UPA scheme against man-in-the-middle (MITM) attack. Attackers can capture packets then cut off the client and server communication and send or modify the packet from a client who had been hacked. As a result, the attacker will be able to log into the application. These weakness isn’t caused from mistake in implementation process. The implementation process has been follow the scheme process. These weakness can be found in authentication process point 3.3 in section 2.1.
5. Conclusion
Through the test results using the OWASP-AT-008, can be concluded that the scheme successfully ensures human participation in the process of authentication when these circumstances are fulfilled: the CAPTCHAs that are used unbreakable, All of the parameters required to decode CAPTCHA are secure from attackers, the CAPTCHA implementation is secure against replay attacks, and the encryption algorithm used to secure the CAPTCHA parameter is still safe to use. However, in this study we found that this scheme is weak against MITM attack. This occurs because the value of the decoded CAPTCHA that is sent to the server is not secured, to mitigate this a secure protocol such as TLS should be used when implementing the scheme.

6. References
[1] Syed Idrus S Z, Cherrier E, Rosenberger C and Schwartzmann J J 2013 A review on authentication methods Aust. J. Basic Appl Sci 7(5) 95-107
[2] Srivastava K, Awasthi A K, and Mittal R C 2013 A review on remote user authentication schemes using smart cards Qshine 2013 115 pp 729-749
[31] Chen T and Huang J 2010 A novel user-participating authentication scheme J. Syst Softw 83(5) 861-867
[4] Kaul S D and Awasthi A K 2013 Analysis and improvement of an authentication scheme using smart cards Qshine 2013 115 719-728
[5] Li L H and Hwang M S 2000 A new remote user authentication scheme using smart cards IEEE Transaction Consum. Electron 46 28-30
[6] Zavlaris P 2017 (Bad Bots) If You Build It, They Will Come [Online] Available: https://resources.distilnetworks.com/all-blog-posts/bad-bot-report-2017-intro [Accessed: 08-Jan-2017].
[7] Von Ahn L, Blum M and Langford J 2004 Telling humans and computers apart automatically Commun. ACM 47(2) 56-60
[8] F. OWASP 2008 OWASP Testing Guide v3.0 OWASP Found p 349
[9] Naor M and Shamir A 1995 Visual cryptography: Advances in Cryptology EUROCRYPT’94 950 1-12
[10] Mamun A A, Rahman S S M and Shao T A 2017 Security analysis of AES and enhancing its security by modifying s-box with an additional byte Int. J. Comput Networks Communications 9(2) 69-88
[11] Gupta S, Yadav S K and Singh A P 2016 A Comparative study of secure hash algorithms in Proc. of First Int.Conf. on Information and Communication Technology for Intelligent Systems 2 125-133