Applied Mathematical Sciences, Vol. 9, 2015, no. 145, 7237 - 7245
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2015.510647

Cryptanalysis of Lightweight Authentication Scheme with User Untraceability

Hae-Jung Kim

College of Liberal Education, Keimyung University
Daegu 704-701, Republic of Korea

Eun-Jun Yoon[1]

Department of Computer Engineering, Kyungil University
Kyungsangpuk-Do 712-701, Republic of Korea

Abstract

In 2015, Yeh proposed a lightweight authentication scheme with user untraceability. Yeh claimed that their proposed scheme is secure to an off-line password guessing attack. However, this paper points out that Yeh's scheme has the following security and efficiency problems unlike their claims: (1) The scheme is still vulnerable to off-line password guessing attack with stolen smart card. (2) The scheme has computational efficiency problem when the service provider authenticates the user sending message. For this reason, Yeh's scheme is insecure for practical application.

Keywords: Cryptography; Lightweight authentication; Smart cards; Cryptanalysis; Password guessing attack; Efficiency problem

1 Introduction

A password-based authentication scheme provides an efficient and accurate way to identify valid remote users. Up to now, many password-based au-

[1]Corresponding author. Tel.: +82-53-600-5623; Fax: +82-53-600-5639

thentication schemes have been investigated in recent years for preserving secrecy of communication[1, 2, 3, 4]. Recently, Chang et al.[3] first introduced a formally provable secure authentication protocol with the property of user-untraceability. Unfortunately, Yeh[4] pointed out that Chang et al.'s scheme fails to provide the property of user-untraceability and is insecure against user impersonation attack, server counterfeit attack, and man-in-the-middle attack. Yeh also proposed a security enhanced authentication scheme to eliminate all identified weaknesses of Chang et al.'s scheme.

However, this paper points out that Yeh's scheme has the following security and efficiency problems[5, 6] unlike their claims: (1) The scheme is still vulnerable to off-line password guessing attack with stolen smart card[6]. (2) The scheme has computational efficiency problem when the service provider authenticates the user sending message[5]. For this reason, Yeh's scheme is insecure for practical application.

This paper is organized as follows: Section 2 briefly reviews the Yeh's lightweight authentication scheme with user untraceability. The security flaws of Yeh's scheme are shown in Section 3. Finally, conclusions are given in Section 4.

2 Review of Yeh's Authentication Scheme

This section briefly reviews Yeh's lightweight authentication scheme with user untraceability[4]. There are three phases in the Yeh's scheme: user registration phase, user login and authentication phase, and password change phase. We outlined some notations used in this research paper.

- U_i : Legitimate user
- S : Service provider
- ID_i, PW_i : U_i's identity and password
- CID_i : U_i's dynamic identity
- $h(\cdot)$: A secure one-way hash function, such as SHA-2 (256 to 512 bits)
- x : A secret key of S
- y : A secret number of S
- $\oplus$: Bit-wise exclusive or operation
- $||$: Concatenate operation

2.1 User Registration Phase

The user registration phase performs as follows:

1. $U_i \rightarrow S$: $\{ID_i, PW_i, r_1, r_2\}$

   When a user U_i wants to access the service of S, U_i chooses and sends his/her identity ID_i, password PW_i, and two chosen random numbers r_1 and r_2 to S via a secure channel.

2. $S \rightarrow U_i$: Smart card containing $\{N_i, r_1, M_i, h(\cdot)\}$

   After S receives the registration request, S computes the followings:

$$M_i = h(y||r_2) \oplus h(PW_i||r_1) \tag{1}$$

$$N_i = h(ID_i||x) \oplus h(PW_i||r_1) \tag{2}$$

   Finally, S stores parameters $\{N_i, r_1, M_i, h(\cdot)\}$ into U_i's smart card and issues this smart card to U_i securely.

3. At the same time, S stores $h(h(y||r_2))$ without any information connected to U_i. That is, S first deletes the registration information related to U_i, and then maintains the secret value $h(h(y||r_2))$ as a random number in a pre-defined table T; hence, S cannot recognize U_i via $h(h(y||r_2))$ or any other information from this table.

2.2 Login and Authentication Phase

Assume that a user U_i wants to login and authenticate to the service provider S. Fig. 1 depicts the Yeh's user login and authentication phase and it performs as follows:

1. $U_i \rightarrow S$: $\{D, CID_i, N_i', C\}$

   When U_i intends to access S, U_i inserts his/her smart card into a card reader, and inputs ID_i and PW_i. The smart card then generates a robust strongly random number r_3 and then computes the followings:

$$A = M_i \oplus h(PW_i||r_1) = h(y||r_2) \tag{3}$$

$$D = r_3 \oplus h(A) \tag{4}$$

$$B = N_i \oplus h(PW_i||r_1) = h(ID_i||x) \tag{5}$$

$$N_i' = N_i \oplus h(h(A)||r_3) \tag{6}$$

$$CID_i = ID_i \oplus h(N_i||h(A)||r_3) \tag{7}$$

$$C = h(N_i||h(A)||B||r_3) \tag{8}$$

Finally, the smart card sends $\{D, CID_i, N_i', C\}$ to S through a public channel.

2. $S \rightarrow U_i$: $\{a, r_4\}$

Once S gets $\{D, CID_i, N_i', C\}$, S iteratively retrieves each value $h(h(y||r_2))^*$ from the maintained table T, and then computes the followings:

$$r_3^* = D \oplus h(h(y||r_2))^* \tag{9}$$

$$N_i^* = N_i' \oplus h(h(h(y||r_2))^*||r_3^*) \tag{10}$$

$$ID_i^* = CID_i \oplus h(N_i^*||h(h(y||r_2))^*||r_3^*) \tag{11}$$

$$B^* = h(ID_i^*||x) \tag{12}$$

$$C^* = h(N_i^*||h(h(y||r_2))^*||B^*||r_3*) \tag{13}$$

After that, S checks if C^* equals C. If equal, U_i is successfully authenticated. Next, S examines r_3^*, and if no login request with the same parameters $\{D, CID_i, N_i', C\}$ is received within a predefined time interval. Once both conditions hold, this phase is passed. Otherwise, S immediately terminates this phase. Note that a threshold value of the number of cumulative failed requests can be set to 3. Then, S generates a random number r_4 and computes a as follows:

$$a = h(B^*||h(h(y||r_2))^*||r_4) \tag{14}$$

Finally, S sends $\{a, r_4\}$ to the smart card via a common channel.

3. Upon receiving $\{a, r_4\}$ from S, the smart card checks the freshness of r_4. If r_4 is fresh in an expected time interval, the smart card computes a^* as follows:

$$a^* = h(B||h(A)||r_4) \tag{15}$$

and compares a^* with a. If values a^* and a are the same, U_i authenticates S.

2.3 Password Change Phase

The user password change phase performs as follows:

1. When U_i wants to change the password, U_i inserts the smart card into the card reader and keys in his/her identity ID_i, password PW_i, and a new password PW_i^{new}.

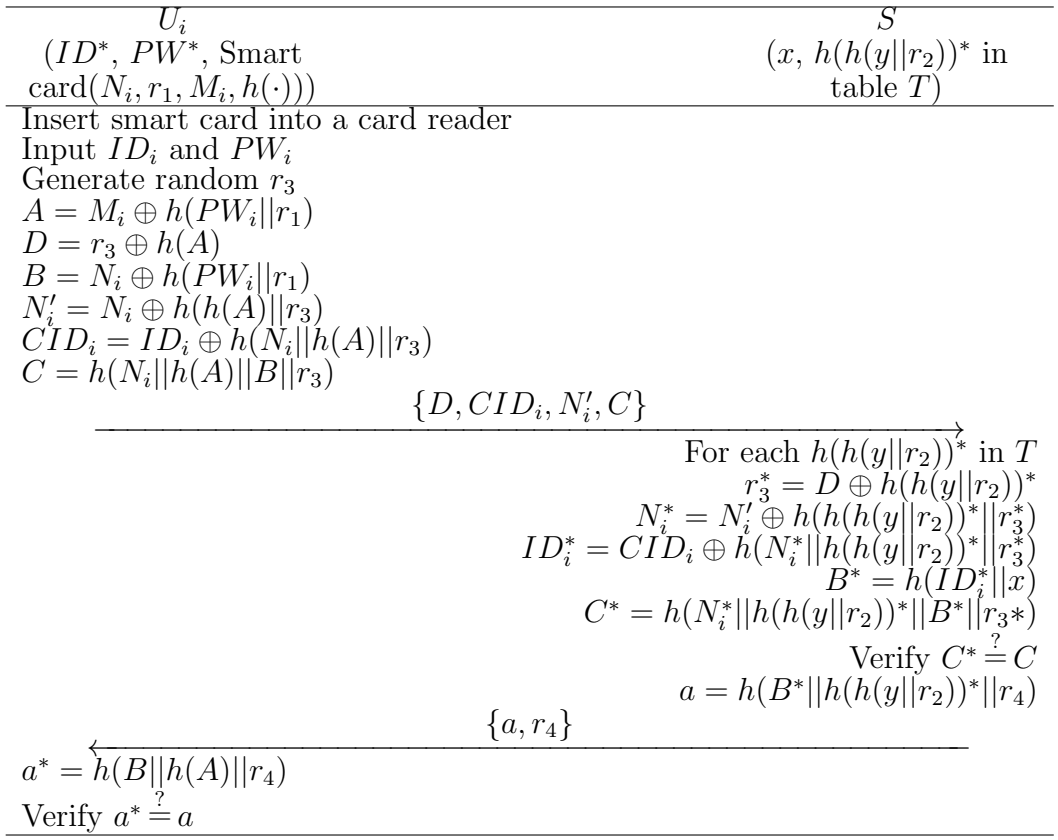

Figure 1: Login and Authentication Phase of Yeh's Scheme

2. The smart card calculates values M_i^{new} and N_i^{new} as follows:

$$M_i^{new} = M_i \oplus h(PW_i||r_1) \oplus h(PW_i^{new}||r_1) \tag{16}$$

$$N_i^{new} = N_i \oplus h(PW_i||r_1) \oplus h(PW_i^{new}||r_1) \tag{17}$$

The smart card stores M_i^{new} and N_i^{new} in the smart card's memory.

Note that a threshold value of the number of cumulative failed requests can be set to 3. If the number of failed requests exceeds 3, the smart card will be locked and only a specific user reverification procedure can be used to unlock this smart card. This mechanism can be exploited to be secure against an off-line password guessing attack as the number of instances of password guessing and testing is limited.

3 Cryptanalysis of Yeh's Authentication Scheme

This section demonstrates that Yeh's lightweight authentication scheme with user untraceability[4] cannot withstand an off-line password guessing attack with stolen smart card[6] and has computational efficiency problem[5].

3.1 Off-line Password Guessing Attack

An off-line password guessing attack is the most powerful attack to the password-based authentication schemes. In the off-line password guessing attack, an attacker uses a guessed password to verify the correctness of the password in an off-line manner[6]. The attacker can freely guess a password and then check if it is correct without limitation in the number of guesses.

Suppose that the attacker steals the parameters $\{N_i, r_1, M_i, h(\cdot)\}$ from a lost smart card and knows the server S's response message $\{a, r_4\}$. Then the attacker can perform an off-line password guessing attack to obtain the password PW_i of the user as follows:

1. The attacker selects a candidate password PW_i^*.
2. The attacker computes $B^* = N_i \oplus h(PW_i^*||r_1)$.
3. The attacker computes $A^* = M_i \oplus h(PW_i^*||r_1)$.
4. The attacker computes $a^* = h(B^*||h(A^*)||r_4)$.
5. The attacker checks if the following equation holds or not

$$a \stackrel{?}{=} a^* \tag{18}$$

   If the check passes, then the attacker confirms that the guessed password PW_i^* is the correct one.

6. If it is not correct, the attacker chooses another password PW_i^{**} and repeatedly performs above steps (2)$\sim$(5) until $a \stackrel{?}{=} a^{**}$.

It is clear that if $PW_i^* = PW_i$, then

$$\begin{aligned} a^* &= h(B^*||h(A^*)||r_4) \\ &= h(N_i \oplus h(PW_i^*||r_1)||h(M_i \oplus h(PW_i^*||r_1))||r_4) \\ &= h(h(ID_i||x) \oplus h(PW_i||r_1) \oplus h(PW_i^*||r_1)||h(h(y||r_2) \oplus h(PW_i||r_1) \oplus h(PW_i^*||r_1))||r_4) \\ &= h(h(ID_i||x)||h(h(y||r_2))||r_4) \\ &= h(B||h(A)||r_4) \\ &= a \end{aligned} \tag{19}$$

Therefore, Yeh's scheme is vulnerable to the off-line password guessing attack. The algorithm of the off-line password guessing attack for getting the password PW_i^* is as follows:

Password Guessing Attack$(N_i, r_1, M_i, a, r_4, h(\cdot), \mathbb{D}_{PW}))$
{
  for $i := 0$ to $|\mathbb{D}_{PW}|$
  {
    $PW_i^* \leftarrow \mathbb{D}_{PW}$;
    $B^* = N_i \oplus h(PW_i^*||r_1)$;
    $A^* = M_i \oplus h(PW_i^*||r_1)$;
    $a^* = h(B^*||h(A^*)||r_4)$;
    if $a \stackrel{?}{=} a^*$ **then**
      return PW_i^*
  }
}

The running time of the above off-line password guessing attack is $(O(|\mathbb{D}_{PW}|) \times 2T_X \times 3T_H)$, where T_X and T_H represent the execution time of bit-wise XOR operations and hash operations respectively. The search spaces $\mathbb{D}_{PW}$ is unlikely to be large enough (for example, $|\mathbb{D}_{PW}| \leq 10^6$), and the time complexities T_X and T_H all can be executed with negligible amount of time, thus the polynomial time-bounded attacker can find the exact password PW of the user easily[6].

3.2 Computational Efficiency Problems

We can see that Yeh's scheme has computational efficiency problems when the service provider S verifies the user U_i's sending message[5]. That is, S must spend heavy verification computation times based on the received message to find the registered user U_i from his/her user account table T.

To provide anonymity of the user U_i in the Yeh's scheme, the identity ID_i of U_i is encrypted by using the shared secret value $A = h(y||r_2)$ between U_i and S as $CID_i = ID_i \oplus h(N_i||h(A)||r_3)$ in step (1). Upon receiving the message CID_i in step (2), S will decrypt it with the stored all secret value $A = h(y||r_2)$ in T to obtain U_i's identity ID_i, where $1 \leq i \leq n$ and n is the number of registered secret values in S. For $1 \leq i \leq n$, S first will decrypt CID_i by using N_i, $A = h(y||r_2)$, and r_3 to obtain ID_i. And then S will verify whether ID_i is the registered user U_i from his/her user account table T by comparing $C^* \stackrel{?}{=} C$. Therefore, S must perform four hash operations for each stored $h(y||r_2)^*$ in T. In this case, for n registered $h(y||r_2)^*$ in T, the maximum time complexity is $O(4n)$ because the hash time complexity is $O(4n)$.

Moreover, if an attacker modifies the message $\{D, CID_i, N_i', C\}$ with a random message of same bit size and then sends it to S, the S cannot verify a legal identity ID_i from his/her user accoun table T. Then, S will reject the forged message after performing $O(4n)$ computation times. In this case, huge resource exhaustion problem will be occurred in S. Moreover, when the attacker sends n random messages to S, the S must perform $O(4(n^2))$ computation times. Based on the above efficiency analysis, we can conclude that Yeh's scheme has computational efficiency problems because it has problems of large storage requirement and corresponding verification overheads.

4 Conclusions

This paper reviewed Yeh's lightweight authentication scheme with user untraceability and then pointed out that Yeh's scheme is still vulnerable to offline password guessing attack using stolen smart card unlike their claims and it has computational efficiency problem when the service provider authenticates the user sending message. For this reason, Yeh's scheme is insecure for practical application. Further works will be focused on improving the Yeh's scheme which can be able to provide greater security and to be more efficient than the existing lightweight authentication scheme with user untraceability by an accurate performance analysis.

Acknowledgements. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIP) (No. 2015R1A2A2A01006824).

References

[1] J. Tsai, N. Lo and T. Wu, Novel anonymous authentication scheme using smart cards, *IEEE Trans. Ind. Inform.*, **9** (2013), no. 4, 2004-2013. http://dx.doi.org/10.1109/tii.2012.2230639

[2] S. Kumari and M. Khan, Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme, *Int. J. Commun. Syst.*, **27** (2013), no. 12, 3939-3955. http://dx.doi.org/10.1002/dac.2590

[3] Y. Chang, W. Tai and H. Chang, Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update, *Int. J. Commun. Syst.*, **27** (2013), no. 11, 3430-3440. http://dx.doi.org/10.1002/dac.2552

[4] K. Yeh, A lightweight authentication scheme with user untraceability, *Front. Inform. Technol. Electron. Eng.*, **16** (2015), no. 4, 259-271. http://dx.doi.org/10.1631/fitee.1400232

[5] E. Yoon, Efficiency and security problems of anonymous key agreement protocol based on chaotic maps, *Commun. Nonlinear. Sci. Numer. Simulat.*, **17** (2012), no. 1, 2735-2740. http://dx.doi.org/10.1016/j.cnsns.2011.11.010

[6] S. Choi and E. Yoon, Cryptanalysis of Tso et al.'s password authentication scheme based on smart card, *Applied Mathematical Sciences*, **9** (2015), no. 101, 5027-5034. http://dx.doi.org/10.12988/ams.2015.56420

Received: October 30, 2015; Published: December 15, 2015