Cryptanalysis of Two-party Explicit Authenticated Key Agreement Protocol

Eun-Jun Yoon\textsuperscript{1}

Department of Cyber Security, Kyungil University
33 Buho-Ri, Hayang-Ub, Kyungsan-Si
Kyungsangpuk-Do 712-701, Republic of Korea

Copyright © 2015 Eun-Jun Yoon. This article is distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In 2015, Zheng et al. proposed an efficient protocol for two-party explicit authenticated key agreement (2EAKA). Zheng et al. claimed that their proposed 2EAKA protocol does not need any fixed public key infrastructure and is provably secure in the random oracle under the Computation Gap Diffie-Hellman assumption. However, this paper points out that Zheng et al.’s 2EAKA protocol still suffers from impersonation attacks based on off-line password guessing attack.

Keywords: Cryptography; Password-based authentication; Explicit authentication; Key agreement; Bilinear pairing; Impersonation attack

1 Introduction

Two-party key agreement protocols\cite{1, 2, 3, 4, 5, 6, 7} enable two honest communication parties (for example, users $A$ and $B$) to communicate over an open insecure network in order to establish a shared common secret session key which can be used later to achieve the cryptographic goals such as confidentiality, integrity and authenticity\cite{7}.

The two-party key agreement protocol can be classified into two categories: implicit key authentication and explicit key authentication\cite{1, 2, 7}. A two-party key agreement protocol provides implicit key authentication if entity $A$ is

\textsuperscript{1}Corresponding author: Eun-Jun Yoon, Tel.: +82-53-600-5623; Fax: +82-53-600-5639
assured that no other entity besides entity $B$ can learn the value of a particular secret key. Key conrmation means that if entity $A$ is assured that the other entity $B$ is in possession of the secret key. A two-party key agreement protocol provides explicit key authentication if both implicit key authentication and key conformation are provided for all participating entities.

In 2015, Zheng et al.[7] proposed an efficient protocol for two-party explicit authenticated key agreement(2EAKA). Zheng et al. claimed that their proposed 2EAKA protocol does not need any fixed public key infrastructure and is provably secure in the random oracle under the Computation Gap Diffie-Hellman assumption[8, 9]. However, this paper points out that Zheng et al.’s 2EAKA protocol still suffers from impersonation attacks based on off-line password guessing attack[10, 11, 12, 13]. For this reason, Zheng et al.’s 2EAKA protocol is insecure for practical application.

This paper is organized as follows: Section 2 briefly reviews the Zheng et al.’s 2EAKA protocol. The security flaws of Zheng et al.’s 2EAKA protocol are shown in Section 3. Finally, conclusions are given in Section 4.

2 Review of Zheng et al.’s 2EAKA Protocol

This section briefly reviews Zheng et al.’s 2EAKA protocol[7]. We outlined some notations used in this research paper.

- $ID_A, ID_B$ : Identities of user $A$ and user $B$
- $pw_{AB}$ : Shared password of $A$ and user $B$
- $p, q$ : Secure large prime numbers such that $p = 2 \mod 3$ and $p = 6q - 1$
- $E(F_p)$ : Super singular elliptic curve defined by $y^2 = x^3 + 1$ over $F_p$
- $P$ : Primitive generator for $E(F_p)$ with order $q$
- $G_1$ : Cyclic additive subgroup of $E(F_p)$ generated by the base point $P$
- $G_2$ : Cyclic additive subgroup of $F_q^2$ order $q$
- $\hat{e}(\cdot) :$ An admissible pairing $\hat{e} : G_1 \times G_1 \rightarrow G_2$
- $H(\cdot) :$ A secure one-way collision-free hash function
- $F(\cdot) :$ A one-way hash function that maps a string to an element of $G_1$

In the Zheng et al.’s 2EAKA protocol, $A$ and $B$ establish a secure high-entropy session key $sk$ from a shared human-memorable password $pw_{AB}$ over a public insecure network under the active adversary. Fig. 1 depicts the Zheng et al.’s 2EAKA protocol, which works as follows:
Cryptanalysis of two-party explicit authenticated key agreement protocol

Choose random $a \in Z_p^*$

$\eta_a = aP + pw_{AB}P$ \hspace{1cm} (1)

Finally, A sends $\{ID_A, \eta_a\}$ to user B.

Step 2. $B \rightarrow A$: $\{ID_B, \eta_b, \mu_b\}$

Upon receiving the message $\{ID_A, \eta_a\}$, user B also selects a random element $b \in Z_p^*$, and then computes the followings:

$aP = \eta_a - pw_{AB}P$ \hspace{1cm} (2)

$R = F(ID_A, ID_B)$ \hspace{1cm} (3)

Session key $sk = H(ID_A, ID_B, k)$
Finally, $B$ sends $\{ID_B, \eta_b, \mu_b\}$ to user $A$.

**Step 3. $A \rightarrow B$: $\{\mu_a\}$**

Upon receiving the message $\{ID_B, \eta_b, \mu_b\}$, $A$ computes the followings:

\[
s = \hat{e}(6aP, 6bR) = \hat{e}(6P, 6R)^{ab}
\]

\[
k = H(aP, bP, R, s)
\]

\[
\mu_b = H(ID_B, k)
\]

\[
\eta_b = bP + pw_{AB}P
\]

and then checks whether the equality

\[
\mu_b \overset{?}{=} H(ID_B, k)
\]

holds or not. If it does not hold, user $A$ terminates this session. Otherwise, $A$ computes the followings:

\[
sk = H(ID_A, ID_B, k)
\]

\[
\mu_a = H(ID_A, k)
\]

Finally, $A$ sends $\{\mu_a\}$ to user $B$.

**Step 4. Upon receiving the message $\{\mu_a\}$, user $B$ checks whether the equality**

\[
\mu_a \overset{?}{=} H(ID_A, k)
\]

**holds or not. If it holds, user $B$ generates the session key**

\[
sk = H(ID_A, ID_B, k)
\]

Otherwise, $B$ terminates this session.

Finally, $A$ and $B$ agree on the common session key $sk = H(ID_A, ID_B, k)$. Both sides will agree on the session key $sk$ if all communication steps are executed correctly. Once the Zheng et al.’s 2EAKA protocol run completes successfully, both parties may use $sk$ to encrypt their subsequent session traffic in order to create a confidential communication channel.
3 Cryptanalysis of Zheng et al.’s 2EAKA Protocol

This section demonstrates that Zheng et al.’s 2EAKA protocol is vulnerable to impersonation attacks. Here, we show that an adversary Eve could obtain the correct shared password by masquerading as the legitimate user A and doing the following steps:

A1. Eve picks a random element $e \in Z_p^*$, then computes $\eta_e = eP$. Finally, Eve sends $\{ID_A, \eta_e\}$ to user B.

A2. Upon receiving the messages $\{ID_A, \eta_e\}$, user B will select a random element $b \in Z_p^*$, and then compute the followings:

$$eP - pw_{AB}P = \eta_e - pw_{AB}P$$

$$R = F(ID_A, ID_B)$$

$$s = \hat{e}(6(eP - pw_{AB}P), 6bR) = \hat{e}(6P, 6R)^{(e - pw_{AB})b}$$

$$k = H(eP - pw_{AB}P, bP, R, s) = H((e - pw_{AB})P, bP, R, s)$$

$$\mu_b = H(ID_B, k)$$

$$\eta_b = bP + pw_{AB}P$$

Finally, B will send $\{ID_B, \eta_b, \mu_b\}$ to the adversary Eve.

A3. Upon receiving the messages $\{ID_B, \eta_b, \mu_b\}$, Eve can compute the correct shared password $pw_{AB}$ by the following off-line password guessing attack:

(a) Select a candidate password $pw_{AB}^*$.  
(b) Compute the followings:

$$R = F(ID_A, ID_B)$$

$$bP = \eta_b - pw_{AB}^*P$$

$$s^* = \hat{e}(6bP, 6(e - pw_{AB}^*)R)$$

$$k^* = H((e - pw_{AB}^*)P, bP, R, s^*)$$

(c) Checks if the following equation holds or not

$$\mu_b \overset{?}{=} H(ID_B, k^*)$$

If the check passes, then Eve confirms that the guessed password $pw_{AB}^*$ is the correct one.
It is clear that if $pw_{AB}^* \equiv pw_{AB}$, then
\[
H(ID_B, k^*) = H(ID_B, H((e - pw_{AB}^*)P, bP, R, s^*)) \\
= H(ID_B, H((e - pw_{AB}^*)P, bP, R, \hat{e}(6bP, 6(e - pw_{AB}^*)R))) \\
= H(ID_B, H((e - pw_{AB}^*)P, bP, R, \hat{e}(6P, 6R)(e - pw_{AB}^*)b)) \\
= H(ID_B, H((e - pw_{AB})P, bP, R, s)) \\
= H(ID_B, H(eP - pw_{AB}P, bP, R, s)) \\
= H(ID_B, k) \\
= \mu_b
\] (28)

(d) If it is not correct, Eve chooses another password $pw_{AB}^{**}$ and repeatedly performs above steps (b) and (c) until
\[
\mu_b \nmid H(ID_B, k^{**})
\] (29)

A4. By using the guessed password $pw_{AB}^*$, the attacker Eve can freely perform the user impersonation attack or the server impersonation attack. Therefore, Zheng et al.’s 2EAKA protocol is vulnerable to the above impersonation attacks based on off-line password guessing attack[10, 11, 12, 13].

4 Conclusions

This paper reviewed Zheng et al.’s 2EAKA protocol and then pointed out that the 2EAKA protocol is still vulnerable to impersonation attacks based on off-line password guessing attack unlike their claims. Consequently, Zheng et al.’s 2EAKA protocol is insecure for practical application. Further works will be focused on improving the 2EAKA protocol which can be able to provide greater security and to be more efficient than the existing password-based explicit authenticated key agreement protocols 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) and partially supported by Small & Medium Business Administration grant funded by the Korea government(SMBA)(No. C0248233).

References


Received: September 29, 2015; Published: December 8, 2015