Comments on a Provably Secure Three-Party Password-Based Authenticated Key Exchange Protocol Using Weil Pairings

Hung-Yu Chien

Department of Information Management

ChaoYang University of Technology , Taiwan, R.O.C.

hychien@cyut.edu.tw

Abstract

In 2005, Wen et al. proposed the first provably secure three-party password-based authenticated key exchange using Weil pairings, and provided their proof in a modified Bellare-Rogaway model (BR-model). Here, we show an impersonation attack on Wen et al.'s scheme and point out a main flaw of their model that allows a man-in-the-middle adversary easily violate the security.

Keywords: bilinear pairings, authenticated key exchange, random oracle model.

1. Introduction

To avoid the inconvenience of key management of two-party password-based authenticated key exchange (two-PAKE) protocols, Wen et al. [1] proposed a provably secure three-party password-based authenticated key exchange (three-PAKE) protocol, using Weil pairings. The three-party protocol requires each entity pre-share a password with a trusted server. Thus, any two entities can mutually authenticate each other and establish a secure session key through the server's assistance. They provided their proof of the protocol in their modified Bellare-Rogaway model [2-4]. Unfortunately, this article will show an impersonation attack on the protocol and point out the main flaws in their modified model.

2. Review of Wen et al.'s protocol

Wen et al. proposed their three-PAKE protocol, using the modified Weil pairings [5]. They provided a security proof of their protocol relative to the Bilinear Diffie-Hellman problem (which is called the Weil Diffie-Hellman assumption in Wen et al.'s paper) in their modified model. Here, to clearly and concisely present their protocol, we introduce the protocol using the general bilinear pairings.

Bilinear pairing: Let G_1 be an additive group of prime order q and G_2 a cyclic multiplicative group of the same order q. The discrete logarithm problems (DLP) in both G_1 and G_2 are assumed to be hard. Let P be a generator of G_1. $H:\{0,1\}^* \to Z_q$ be one cryptographic hash functions, and G: $\{0,1\}^* \to G_1$ be the cryptographic one-way hash function that maps a string to a point of G_1 [5]. $\hat{e}$: $G_1 \mathrm{x} G_1 \rightarrow G_2$ be a bilinear mapping satisfying the following conditions.

1. Bilinear：Let $a,b \in Z$ and $P,Q \in G_1$, $\hat{e}(a \cdot P, b \cdot Q) = \hat{e}(P,Q)^{ab}$.
2. Non-degenerate：There exists $P \in G_1$ such that $\hat{e}(P,P) \neq 1 \in G_2$.
3. Polynomial-time computable：The mapping function $\hat{e}(P,Q)$ is computable in polynomial time.

Bilinear Diffie-Hellman Problem (BDHP) for a bilinear pairing $\hat{e}$: $G_1 \mathrm{x} G_1 \rightarrow G_2$ is defined as follows: Given $P, aP, bP, cP \in G_1$, compute $\hat{e}(P,P)^{abc}$, where a,b,c are random numbers from Z_q^*. It is commonly believed that the BDHP problem is hard.

Wen et al.'s three-party PAKE protocol

Setup: Let (G_1, G_2, *H, G,* $\hat{e}$(), *E*()/*D*()) be the public system parameters, where *E*() denotes an ideal symmetric encryption function and *D*() denotes the

corresponding decryption function. ID_S/ID_A/ID_B respectively denotes the identity of the authentication server *S*/user *A*/user *B*. The server *S* owns its secret key *s* and publicizes its public key $P_S = sP$. The users *A* and *B* share passwords PW_A and PW_B with the server *S*, respectively.

Execution: To share an authenticated session key, the server *S*, the users *A* and *B* perform the following steps. "*A* → *B*: *M*" denotes that *A* sends the message *M* to *B*.

Step 1. *A* → *B*: (ID_A, aP, c_a)

User *A* selects a random number *a*, computes *aP* and $k_a = H(aP, P_S, Q, \hat{e}(P_S, aQ))$, where $Q = G(ID_S)$. Then *A* computes $c_a = E_{k_a}(PW_A)$ and sends (ID_A, aP, c_a) to user *B*.

Step 2. *B*→ *S*: $(ID_A, aP, c_a, ID_B, bP, c_b, \mu_b)$[1]

User *B* randomly selects an integer *b*, computes *bP*, $k_b = H(bP, P_S, Q, \hat{e}(P_S, bQ))$ and $K = \hat{e}(aP, bU)$, where $U = G(ID_A, ID_B)$. Then B computes $c_b = E_{k_b}(PW_B)$ and $\mu_b = H(ID_B, K)$. Finally, *B* sends $(ID_A, aP, c_a, ID_B, bP, c_b, \mu_b)$ to server *S*.

Step 3. *S* → *A*: $(ID_B, bP, \mu_b, \sigma_b, \sigma_a)$

S computes $k_a = H(aP, P_S, Q, \hat{e}(aP, sQ))$ and $k_b = H(bP, P_S, Q, \hat{e}(bP, sQ))$, and verifies the equality $PW_A \stackrel{?}{=} D_{k_a}(c_a)$ and $PW_B \stackrel{?}{=} D_{k_b}(c_b)$, respectively. If any one of the verifications fails, *S* rejects the session; otherwise, *S* computes $\sigma_a = H(k_b, aP)$ and $\sigma_b = H(k_a, bP)$, and sends $(bP, \mu_b, \sigma_b, \sigma_a)$ to user *A*.

Step 4. *A* → *B*: (μ_a, σ_a)

A computes $K = e(bP, aU)$ and checks the equality of $\sigma_b \stackrel{?}{=} H(k_a, bP)$ and $\mu_b \stackrel{?}{=} H(ID_B, K)$, respectively. If any one of the verifications fails, *A* rejects the

[1] Wen et al. did not specify the identities of communicating parties in Step 2-4, maybe due to typing errors. Here, we explicitly add the identities of the entities.

session. Otherwise, *A* computes $\mu_a = H(ID_A, K)$ and sends (μ_a, σ_a) to *B*. Upon receiving the data in Step 4, *B* verifies the equality $\sigma_a \stackrel{?}{=} H(k_b, aP)$ and $\mu_a \stackrel{?}{=} H(ID_A, K)$, respectively. If any verification fails, *B* rejects the session; otherwise, *B* accepts and completes the session. The final session key shared between *A* and *B* is $SK = H(aP, bP, U, K)$.

3. Impersonation attack on Wen et al.'s scheme and flaws of Wen et al.'s model

This section first shows an impersonation attack on Wen et al.'s scheme, and then point out the main flaw in their model.

3.1 Impersonation attack

Now we demonstrate an adversary, say user *E* who owns his identity ID_E and shares his password PW_E with the server *S*, can easily impersonate *B* to complete the protocol and share a session key with *A* as follows. In the following scenario, the notation "*B*(*E*)" denotes that *E* impersonate *B* to send messages. "$A \stackrel{\not\rightarrow}{E} B$" denotes that *A* sends messages to *B*, but the messages are intercepted by *E*.

Step 1. $A \stackrel{\not\rightarrow}{E} B$: (ID_A, aP, c_a)

Like the normal interaction, *A* sends *B* the message (ID_A, aP, c_a), but this message is intercepted by *E*.

Step 2. $E \rightarrow S$: $(ID_A, aP, c_a, ID_E, eP, c_e, \mu_e)$

Adversary *E* randomly selects an integer e, computes eP, $k_e = H(eP, P_S, Q, \hat{e}(P_S, eQ))$ and $K = \hat{e}(aP, eU)$, where $U = G(ID_A, ID_B)$. Then *E* computes $c_e = E_{k_e}(PW_E)$ and $\mu_e = H(ID_B, K)$. Finally, *E* sends $(ID_A, aP, c_a, ID_E, eP, c_e, \mu_e)$ to server *S*. Please notice that *E* uses its own identity

ID_E and its password PW_E to compute c_e such that the server *S* will later successfully authenticate *E* as the identity claims; however, *E* uses *aP*, *eP*, ID_A and ID_B to compute the values *K* and *U* such that *A* later will share the same key with *E*.

Step 3. $S \underset{E}{\not\rightarrow} A$: $(ID_E, eP, \mu_e, \sigma_b, \sigma_a)$

S computes $k_a = H(aP, P_S, Q, \hat{e}(aP, sQ))$ and $k_e = H(eP, P_S, Q, \hat{e}(eP, sQ))$, and verifies the equality $PW_A \stackrel{?}{=} D_{k_a}(c_a)$ and $PW_E \stackrel{?}{=} D_{k_e}(c_e)$, respectively. So, the server *S* successfully authenticate *A* and *E*. Then *S* computes $\sigma_a = H(k_e, aP)$ and $\sigma_b = H(k_a, eP)$, and sends $(ID_E, eP, \mu_e, \sigma_b, \sigma_a)$ to user *A*, but this message is intercepted by *E*.

Step 3'. *S*(*E*) → *A*: $(ID_B, eP, \mu_e, \sigma_b, \sigma_a)$

After intercepting the message in Step 3, *E* replaces the identity ID_E with ID_B, and impersonates *S* to send the message $(ID_B, eP, \mu_e, \sigma_b, \sigma_a)$.

Step 4. $A \underset{E}{\not\rightarrow} B$: (μ_a, σ_a)

A computes $K = \hat{e}(eP, aU)$ and checks the equality of $\sigma_b \stackrel{?}{=} H(k_a, eP)$ and $\mu_e \stackrel{?}{=} H(ID_B, K)$, respectively. Since both the verifications succeed, *A* wrongly believes she is communicating with *B*, and will compute $\mu_a = H(ID_A, K)$ and send (μ_a, σ_a) to *B*. This message is also intercepted by *E*. Finally, *A* wrongly believes she is communicating with *B*, but *A* un-intentionally shares the session key $SK = H(aP, eP, U, K)$ with *E*.

Several variants of the attack can be modified from the above scenario. We will not enumerate all the scenarios, but further point out the main flaw in Wen et al.'s security model as follows.

3.2 Main flaw of Wen et al.'s model

Wen et al.'s model is based on those of Bellare-Rogaway model and its variants

[2-4], where the active adversary controls the channels and his capacities are modeled through several oracle queries- S*end*, *Reveal*, *Corrupt* and *Execute* (please refer to [1-4] for the details). The definition of security depends on the notations of partnership of oracles and in-distinguishability, where the partnership is used in the definition of security to restrict the adversary's *Reveal* and *Corrupt* queries to oracles that are not partners of the oracle whose key the adversary is trying to guess. In the BR95 model [3], partnership is defined using a partner function whose purpose is to enable a mapping between two oracles that should share a secret key on completion of the protocol execution. In the BPR2000 model [4], partnership of oracles is defined using SIDs [7]. Choo et al. [5] had pointed out that the specific partner function defined in the security proof of 3PKD is flawed. Here, we point out that the partner function of Wen et al.'s model is also flawed, and it, therefore, invalidates their proof.

Flaws in partnering of Wen et al.'s model: In Wen et al.'s model, two oracles $\prod_A^i$ and $\prod_B^j$ are partnered if the following conditions hold: (1) $\prod_A^i$ and $\prod_B^j$ directly exchange some message flows; (2) $\prod_A^i$ and $\prod_B^j$ hold the same session key *SK*; (3) no oracles besides holds the session key *SK*. The condition (1) is ambiguous, and it even excludes the capturing of several basic attacks. For example, two communicating entities with a man-in-the-middle adversary do not exchange messages directly; therefore, the two entities are not partnered in Wen et al.'s model. So, the adversary can easily send a *Reveal* query of either one of the two entities to derive the session key.

4. Conclusions

In this article, we have shown the impersonation attack on Wen et al.'s

three-PAKE protocol, and have shown a main flaw of their model. The main flaw allow a man-in-the-middle adversary easily derive the session key and violate the security.

References

[1] H.-A.Wen, T.-F. Lee and T. Hwang, "Provably secure three-party password-based authenticated key exchange protocol using Weil pairing", IEE Proc-Commun., 152(2), pp. 138-143, 2005.

[2] M. Bellare, and P. Rogaway, "Entity authentication and key distribution", *CRYPTO 93*, pp. 232–249, 1994.

[3] M. Bellare, and P. Rogaway, "Provably secure session key distribution- the 3 party case", Proc. 27th ACM Symp. on Theory of Computing, pp. 57–66, 1995.

[4] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attack", *EUROCRYPT* 2000, pp. 122–138, 2000.

[5] D. Boneh, and M. Franklin, "Identity-based encryption from the weil pairing", CRYPTO 2001, pp. 213–229, 2001.

[6] K.K.R. Choo, C. Boyd, Y. Hitchcock, M. Greg, "On session identifiers in provably secure protocols", in Fourth Conference on Security in Communication Networks - SCN 2004, LNCS 3352, Springer-Verlag, pp. 352-267.

[7] H.-Y. Chien, "ID-based Tripartite Multiple Key Agreement Protocol facilitating Computer Auditing and Transaction Refereeing", accepted and to be printed in *Journal of Information Management*, 2006.