Security Analysis of Zhao and Gu’s Key Exchange Protocol

Junghyun Nam¹, Juryon Paik², and Dongho Won²**

¹ Department of Computer Science, Konkuk University, 322 Danwol-dong, Chungju-si, Chungcheongbuk-do 380-701, Korea
jhnam@kku.ac.kr

² Department of Computer Engineering, Sungkyunkwan University, 300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do 440-746, Korea
wise96@ece.skku.ac.kr,dhwon@security.re.kr

Abstract. Recently, Zhao and Gu proposed a new protocol for password-based authenticated key exchange in the three-party setting, where each user does not need to remember and manage multiple passwords, but shares only a single password with a trusted server. Zhao and Gu’s protocol was claimed to be provably secure in a formal adversarial model which captures strong corruption. In this paper, we conduct a security analysis on the protocol and thereby show that unlike the claim of provable security, the protocol is not secure against strong corruption. Our result implies that the proof of security for the protocol is not correct.

Keywords: Authentication, key exchange, passwords, strong corruption, ephemeral secrets.

1 Introduction

Recently, Zhao and Gu [6] proposed a three-party PAKE protocol making use of the trapdoor test technique introduced by Cash, Kiltz, and Shoup [3]. Zhao and Gu’s protocol was claimed to be provably secure under the assumption that the hash functions used in the protocol are random oracles. The adversarial model, where security of the protocol is proven, captures the notion of strong corruption by allowing the adversary to ask EphemeralKeyReveal queries. An EphemeralKeyReveal query against a user instance outputs all the ephemeral secrets used by the instance during the protocol execution. Allowing an adversary to ask EphemeralKeyReveal queries models the adversary’s capability to embed a Trojan horse or other form of malicious code into a user’s machine and then obtain all the session-specific information of the victim. Since Zhao and Gu’s protocol is proven secure in a model that allows EphemeralKeyReveal queries, it

⋆ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0026023).
⋆⋆ The corresponding author: Dongho Won (e-mail: dhwon@security.re.kr).
Table 1. The sequence of oracle queries

<table>
<thead>
<tr>
<th>Query</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Execute(\Pi_1^A, \Pi_1^B, \Pi_1^S)</td>
<td>Transcript: M_A, M_B, M_{SA}, M_{SB}</td>
</tr>
<tr>
<td>2 EphemeralKeyReveal(\Pi_1^B)</td>
<td>(b_1, b_2, y, k_B)</td>
</tr>
<tr>
<td>3 SendClient(\Pi_2^A, start:B)</td>
<td>M'_A = \langle X'_1, A'_1, A'_2, c'_A, \omega'_A, ID_A, ID_B \rangle</td>
</tr>
<tr>
<td>4 SendServer(\Pi_2^S, M'_A)</td>
<td>M_{SA} = \langle Y, B_1, B_2, V'<em>B, \omega'</em>{SB}, ID_S \rangle</td>
</tr>
<tr>
<td>5 SendServer(\Pi_2^S, M_B)</td>
<td>M_{SB} = \langle X'_1, A'_1, A'<em>2, V_A, \omega'</em>{SA}, ID_S \rangle</td>
</tr>
<tr>
<td>6 SendClient(\Pi_2^A, M_{SA})</td>
<td>(accept)</td>
</tr>
<tr>
<td>7 Test(\Pi_2^A)</td>
<td>SK^2_A or a random key</td>
</tr>
</tbody>
</table>

should be secure against strong corruption. But what we found is the opposite: Zhao and Gu’s protocol does not exhibit resistance against strong corruption. Indeed, Zhao and Gu’s protocol is vulnerable to an ephemeral-key reveal attack where the adversary asks an EphemeralKeyReveal query. We here reveal this security vulnerability of Zhao and Gu’s protocol. Our result invalidates the claimed proof of security for the protocol.

2 Breaking AKE Security

Zhao and Gu’s key exchange protocol [6] comes along with a claimed proof of its security in a formal model of adversarial capabilities. The adversarial model that they used is the one of Yoneyama [5] and captures security against strong corruption [1, 2, 4]. In this section, we break the AKE security of Zhao and Gu’s protocol by mounting an ephemeral-key reveal attack against the protocol. Due to space limitation, we omit the description of the adversarial model; see Section 2 of [6] for the details of the model. Table 1 shows the sequence of oracle queries corresponding to the attack that breaks the AKE security of Zhao and Gu’s protocol. It is easy to see that after all the queries, the instance \Pi_2^A is fresh; (1) no one in \{A, B, S\} has been sent a Corrupt query, (2) no Reveal query has been made against any instance, and (3) the query EphemeralKeyReveal(\Pi_1^B) has been asked before the SendClient queries to \Pi_2^A have been asked. Thus, the adversary C may test (i.e., ask the Test query against) the instance \Pi_2^A, C is able to compute SK^2_A, the session key of \Pi_2^A, on its own since it knows the values of the exponents b_1, b_2, y used to compute B_1, B_2 and Y. This means that C achieves its goal of breaking the AKE security of Zhao and Gu’s protocol.

Generally speaking, it is desirable that ephemeral secrets exposed in a session should not jeopardize the session-key secrecy of any other sessions. For this reason, key exchange protocols proven AKE-secure in a model that allows strong corruption ought to be resistant against any attacks similar to ours. Our attack shows that the proof of security for Zhao and Gu’s protocol is invalid. The problem with the proof is that the result of EphemeralKeyReveal queries was not adequately considered in the simulation.
References


