Security of a Leakage-Resilient Protocol for Key Establishment and Mutual Authentication (Extended Abstract)

Raphael C.-W. Phan[1], Kim-Kwang Raymond Choo[2,⋆], and Swee-Huay Heng[3]

[1] Laboratoire de sécurité et de cryptographie, EPFL, Lausanne, Switzerland
raphael.phan@epfl.ch
[2] Canberra, Australia
raymond.choo.au@gmail.com
[3] Centre for Cryptography and Information Security (CCIS),
Faculty of Information Science and Technology, Multimedia University, Malaysia
shheng@mmu.edu.my

Abstract. We revisit Shin *et al.*'s leakage-resilient password-based authenticated key establishment protocol (LR-AKEP) and the security model used to prove the security of LR-AKEP. By refining the Leak oracle in the security model, we show that LR-AKE (1) can, in fact, achieve a stronger notion of leakage-resilience than initially claimed and (2) also achieve an additional feature of traceability, not previously mentioned.

Keywords: Key establishment, mutual athentication, leakage-resilient.

1 Introduction

Authenticated Key Establishment protocols (AKEPs) allow two parties to share a secret key based on long-term secrets associated with individual entities (typically passwords). Passwords are strings easily memorized by humans and thus of low entropy. Such protocols are especially popular in computationally restricted devices and those requiring interaction with human users. For example, in practical applications, the secrets derived from passwords are stored in some devices (e.g., a table containing hashed values of passwords kept by a trusted server). A fundamental security threat for password-based AKEPs is, unsurprisingly, dictionary attacks due to low entropy of password-based AKEPs.

We revisit the leakage-resilient password-based AKEPs (LR-AKEPs), first proposed by Shin, Kobara and Imai [7] and subsequently extended in [4,8,9,10]. LR-AKEPs, designed to maintain the secrecy of the long-term password even in the case when stored secrets (i.e., functions of the password) are leaked, can be broadly categorised into two families: the Diffie–Hellman-based LR-AKEPs [7,8,9] and the RSA-based LR-AKEPs [4,10].

⋆ The views and opinions expressed in this paper are those of the author and do not reflect those of any organisation with which the author may be affiliated. This research was undertaken in the author's personal capacity.

W. Susilo, J.K. Liu, and Y. Mu. (Eds.): ProvSec 2007, LNCS 4784, pp. 169–177, 2007.

Widely used security models for AKEPs (including password-based AKEPs) include the indistinguishability-based models of Bellare, Pointcheval, and Rogaway [1] model (hereafter referred to as the BPR2000 model) and Canetti and Krawczyk [2][1]. In the BPR2000 model, leakages of established secret session keys and long-term secrets (e.g., private key or password) are considered by allowing the adversary to have access to the Reveal oracle and the Corrupt oracle respectively. To model leakage-resilience, Shin *et al.* [7] introduced an additional Leak oracle that allows the adversary to learn the stored secrets of unrelated sessions.

The focus of this paper is on the Diffie–Hellman-based LR-AKEP published in ASIACRYPT 2003 [7] (hereafter referred to as "the LR-AKE protocol"). A distinct difference between the LR-AKE protocol and latter extensions [8,9] is that only one secret is stored on the client in the latter schemes.

We regard our contributions in this paper to be three-fold:

1. **Revised security model:** We refine the original model used by Shin *et al.* to prove the security of the LR-AKE protocol by splitting the Leak oracle into LeakC and LeakS oracles[2]. By so doing, we are able to define *how many* leakages occur on the server side.
2. **Stronger notion of leakage-resilience than that defined by Shin *et al.*:** Shin *et al.* proved that the LR-AKE protocol is secure when the leaks do not originate from both the client and servers simultaneously. We demonstrate that the LR-AKE protocol can, in fact, provide an *almost* perfect security level.
3. **Notion of traceability not previously mentioned by Shin *et al.*:** We demonstrate that the LR-AKE protocol can provide traceability, which allows us to identify the compromised client or server devices when leakages occur.

2 Revisiting the Leakage-Resilient AKE Protocol of ASIACRYPT 2003

The notation used throughout this paper is as described in Table 1.

The LR-AKE protocol, described in Fig. 1, can be considered a two-party password-based AKE involving a client-server pair where the server is one out of $n-1$ possible servers. The client, C, remembers a chosen password, pw, and stores $n-1$ secret values, h_i $(i = 1, \ldots, n-1)$, derived from pw in C's device. A partial secret value, $h^{p(i).\lambda_i}$ for $1 \leq i \leq n-1$ (not a share) of pw, is registered with each of the $n-1$ servers. This will enable C to establish a session key with any of these servers in subsequent sessions. The underlined values in Fig. 1 represent the stored secrets of the respective client and server. Note that

[1] Interested reader is referred to [3] for a comparison and a discussion of existing security models for AKEPs.

[2] The definitions of oracles A1 through A4 in Section 4 of [9] implicitly split the Leak oracle, thereby distinguishing whether the leakage occurs at the client or at the server.