Information Processing Letters 116 (2016) 33–40

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On security analysis of an after-the-fact leakage resilient key exchange protocol ☆

Zheng Yang [a,*], Shuangqing Li [b]

[a] School of Computer Science and Engineering, Chongqing University of Technology, China
[b] College of Computer Science, Chongqing University, China

ARTICLE INFO

Article history:
Received 2 June 2014
Received in revised form 18 July 2015
Accepted 13 August 2015
Available online 6 September 2015
Communicated by S.M. Yiu

Keywords:
Cryptography
Cryptanalysis
Authenticated key exchange
Two pass
DDH

ABSTRACT

In this paper, we revisit the security result of an authenticated key exchange (AKE) scheme proposed in AsiaCCS'14 by Alawatugoda, Stebila and Boyd (which is referred to as ASB scheme). The ASB scheme is proved to be secure in a new bounded (continuous) after-the-fact leakage extended Canetti–Krawczyk (B(C)AFL-eCK) model without random oracles, where the B(C)AFL-eCK is extended from the eCK model. However we disprove their security results. We first show an attack against ASB scheme in the eCK model. This also implies that the insecurity of ASB scheme in the B(C)AFL-eCK model. Secondly we point out that the security of ASB scheme is incorrectly reduced to DDH assumption. A solution is proposed to fix the problem of ASB scheme with minimum changes, which yields a new ASB' scheme. We prove the eCK security of ASB' in the random oracle model under Gap Diffie–Hellman assumption.

1. Introduction

A protocol without security proof might be susceptible to active attacks. As proofs are invaluable tools in assuring practitioners about the security attributes of protocols. The essential part of security proof is a security reduction that makes use of the adversary breaking the security goals of considered protocol in certain security model, to solve some computational problem believed to be hard. Hence the development of security models has always gained much attention of research community. Up to now, state-of-art security models (e.g. [3,5,6,1]) have been introduced for evaluating the security of Authenticated Key Exchange (AKE) that follow the first formal AKE security model by Bellare and Rogaway [2]. However a natural question is that whether a protocol with security proof can truly provide security properties as it is claimed? Unfortunately, the answer might be negative. Several results (e.g. [4,9,11,7]) have shown that security proofs of protocols might be flawed which may lead the corresponding protocols to be trivial broken. Hence a security proof is useful if and only if it is correct. But specifying correct computational complexity proof for a protocol remains a hard problem.

Most recently, Alawatugoda et al. [1] studied the problem on partial leakage of long-term secret of protocol principal after the session key is established. Two security models are proposed to formulate the bounded after-the-fact leakage and continuous after-the-fact leakage respectively, which are referred to BAFL-eCK model and CAFL-eCK model respectively. We notice that both models are particularly modified from the extended Canetti–Krawczyk (eCK) model [6]. In order to achieve the security in their proposed models, a somewhat generic construction of a two-pass key exchange protocol (which is referred to as ASB scheme) is introduced based on cryptographic building primitives: leakage-resilient signature scheme and leakage-resilient NAXOS like trick from a pair generation indis-

☆ Project Supported by Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant No. KJ1500918).

* Corresponding author.
E-mail address: zheng.yang@rub.de (Z. Yang).

http://dx.doi.org/10.1016/j.ipl.2015.08.006
0020-0190/

tinguishable leakage-resilient public-key cryptosystem. The security of ASB scheme is claimed to be able to reduce to the security of underlying cryptographic building blocks and the Decisional Diffie–Hellman (DDH) hard problem without random oracles. However, in this work, we show that the ASB scheme is actually not secure in the eCK model at all. This also implies the insecurity of ASB scheme in the B(C)AFL-eCK model. The problem here is that the randomized signature value might be exploited by adversary to lead two non-matching sessions to generate the same session key.

A solution (named ASB' scheme) is proposed to circumvent the problem of ASB scheme, that we only change the key derivation function (KDF) via putting all protocol messages into it. However we find out that even the modified ASB' scheme cannot be reduced to DDH assumption. This result also reflects (from another prospective) that the security proof of ASB scheme is incorrect. In order to fix this proof problem, we re-prove the eCK security of ASB' scheme in the random oracle model based on Gap Diffie–Hellman (GDH) assumption [8]. It is not hard to extend our security proof of ASB' in the BAFL-eCK model. We leave it out as future work. We hope our analysis would be helpful for avoiding similar mistakes while proving the security for ASB style protocol in future works.

2. Security model

In [1], Alawatugoda et al. proposed two security models for AKE: bounded after-the-fact leakage eCK (BAFL-eCK) model and continuous after-the-fact leakage eCK (CAFL-eCK) model. We notice that both models are derived from the extended Canetti–Krawczyk (eCK) model [6]. For simplicity, we only review the eCK model here based on the framework [10], that is enough for us to show the insecurity of ASB scheme [1].[1] Since if the protocol is insecure in the eCK model then it implies the insecurity in the B(C)AFL-eCK model.

Execution environment. In the execution environment, we first consider the formalism of at most $\ell \in \mathbb{N}$ honest parties $\{\mathsf{ID}_1, \ldots, \mathsf{ID}_\ell\}$ for $\ell \in \mathbb{N}$ (that may be under attacked), where ID_i ($i \in [\ell]$) is the identity of a party which is chosen uniquely from space $\mathcal{IDS}$. Each identity is associated with a long-term key pair $(sk_{\mathsf{ID}_i}, pk_{\mathsf{ID}_i}) \in (\mathcal{SK}, \mathcal{PK})$ for authentication. Each honest party ID_i is characterized by a collection of oracles $\{\pi_i^s : i \in [\ell], s \in [d]\}$ for $d \in \mathbb{N}$, where all oracles can be run sequentially and concurrently execute the protocol multiple times with different intended communication partners. Oracle π_i^s behaves as party ID_i carrying out a process to execute the s-th protocol instance (session), which has access to the long-term key pair $(sk_{\mathsf{ID}_i}, pk_{\mathsf{ID}_i})$ and to all other public keys. Moreover, we assume each oracle π_i^s maintains a list of independent internal state variables with semantics listed in Table 1.

All those variables of each oracle are initialized with empty string which is denoted by the symbol $\emptyset$ in the following. At some point, each oracle π_i^s may complete the execution always with a decision state $\Phi_i^s \in \{\mathtt{accept}, \mathtt{reject}\}$. Furthermore, we assume that the session key is assigned to the variable K_i^s (such that $K_i^s \neq \emptyset$) iff oracle π_i^s has reached an internal state $\Phi_i^s = \mathtt{accept}$.

Table 1
Internal states of oracles.

Variable	Decryption
Ψ_i^s	storing the identity and public key of its intended communication partner, e.g. $(\mathsf{ID}_j, pk_{\mathsf{ID}_j})$
Φ_i^s	denoting the decision $\Phi_i^s \in \{\mathtt{accept}, \mathtt{reject}\}$
ρ_i^s	denoting the role $\rho_i^s \in \{Initiator(I), Responder(R)\}$
K_i^s	recording the session key $K_i^s \in \mathcal{K}_{\mathsf{AKE}}$
st_i^s	storing the ephemeral keys that allows to be revealed, e.g. the randomness used to generate ephemeral public key
rT_j^s	recording the transcript of messages received by oracle π_i^s

Adversarial model. An adversary $\mathcal{M}$ in our model is a PPT Turing Machine taking as input the security parameter 1^κ and the public information (e.g. generic description of above environment), which may interact with these oracles by issuing the following queries.

- $\mathsf{Send}(\pi_i^s, m)$: The adversary can use this query to send any message m of his own choice to oracle π_i^s. The oracle will respond the next message m^* (if any) to be sent according to the protocol specification and its internal states. Oracle π_i^s would be initiated as *initiator* via sending the oracle the first message $m = (\top, \widetilde{\mathsf{ID}_j})$ consisting of a special initialization symbol $\top$ and a value $\widetilde{\mathsf{ID}_j}$. The $\widetilde{\mathsf{ID}_j}$ is either the identity ID_j of intended partner or empty string $\emptyset$. After answering a Send query, the variables $(\Psi_i^s, \Phi_i^s, K_i^s, sT_i^s, rT_i^s)$ will be updated depending on the specific protocol.
- $\mathsf{RevealKey}(\pi_i^s)$: Oracle π_i^s responds with the contents of variable K_i^s.
- $\mathsf{EphemeralKeyReveal}(\pi_i^s)$: Oracle π_i^s responds with its ephemeral keys (i.e. per-session randomness of the oracle).
- $\mathsf{Corrupt}(\mathsf{ID}_i)$: The long-term secret key sk_{ID_i} of party ID_i is returned if $i \in [\ell]$; otherwise a failure symbol $\perp$ is returned.
- $\mathsf{Test}(\pi_i^s)$: If the oracle has state $\Phi_i^s = \mathtt{reject}$ or $K_i^s = \emptyset$, then the oracle π_i^s returns some failure symbol $\perp$. Otherwise it flips a fair coin b, samples a random element K_0 from key space $\mathcal{K}_{\mathsf{AKE}}$, and sets $K_1 = K_i^s$. Finally the key K_b is returned.

Secure AKE protocols. In a AKE protocol, two sessions may engage in one partnered on-line communication to establish a session key. This situation is formulated via a notion of *matching sessions*.

Definition 1 *(Matching sessions).* We say that an oracle π_i^s has a *matching session* to oracle π_j^t, if π_i^s has sent all protocol messages and all the following conditions hold:

- $\rho_i^s \neq \rho_j^t$, $\Psi_i^s = (\mathsf{ID}_j, pk_{\mathsf{ID}_j})$, $\Psi_j^t = (\mathsf{ID}_i, pk_{\mathsf{ID}_i})$, and
- $rT_j^t = sT_i^s$ and $rT_i^s = sT_j^t$.

[1] Please note that the BAFL-eCK and CAFL-eCK models are much more complicated than the eCK model.