INFORMATICA, 2006, Vol. 17, No. 4, 467–480

On the Security Analysis of Lee, Hwang & Lee (2004) and Song & Kim (2000) Key Exchange / Agreement Protocols

Kim-Kwang Raymond CHOO [*]
Australian Institute of Criminology
GPO Box 2944, Canberra ACT 2601, Australia
e-mail: raymond.choo@aic.gov.au

Received: May 2005

Abstract. We revisit the password-based group key exchange protocol due to Lee *et al.* (2004), which carries a claimed proof of security in the Bresson *et al.* model under the intractability of the Decisional Diffie–Hellman problem (DDH) and Computational Diffie–Hellman (CDH) problem. We reveal a previously unpublished flaw in the protocol and its proof, whereby we demonstrate that the protocol violates the definition of security in the model. To provide a better insight into the protocol and proof failures, we present a fixed protocol. We hope our analysis will enable similar mistakes to be avoided in the future. We also revisit protocol 4 of Song and Kim (2000), and reveal a previously unpublished flaw in the protocol (i.e., a reflection attack).

Key words: password-based key establishment protocols, key agreement protocols, provable security, information security.

1. Introduction

As the Internet evolves from an academic and research network into a commercial network, more and more organizations and individuals are connecting their internal networks and computers to the insecure Internet. As a result, mass retail electronic commerce in the Internet was born, with more traditional business and services (such as electronic banking, bill payment, gaming) being conducted and offered online over open computer and communications networks.

One of the greatest concerns with this phenomenon is the confidentiality and the integrity of data transmitted over the insecure Internet, and hence the ability to provide security guarantees is of paramount importance. Many initiatives have been proposed to address this concern, which includes cryptographic data encryption and authentication. Typically security guarantees are provided by means of protocols that make use of security primitives such as encryption, digital signatures, and hashing.

[*] The views and opinions expressed in this paper do not necessarily reflect those of the Commonwealth Government, the Minister for Justice and Customs, or the Australian Institute of Criminology. Research was performed while the author was with the Information Security Institute / Queensland University of Technology.

Menezes, van Oorschot, and Vanstone (Menezes *et al.*, 1997; Chapter 1) and Boyd and Mathuria (Boyd and Mathuria, 2003; Chapter 1) identify the following possible different services that may be provided by the employment of cryptographic algorithms.

Confidentiality ensures the data is available only to the authorised parties involved. To achieve this notion, encryption using mathematical algorithms is typically used to encrypt the data and render the encrypted data unintelligible to anyone else, other than the authorised parties even if the unauthorised party (commonly referred to as the adversary in the literature) has access to the encrypted data. In cryptographic protocols, confidentiality ensures that keys and other data are only available to the authorised principals (entities) as intended and trusted third party server if applicable.

Data integrity ensures the data has not been tampered with or modified. To achieve this notion, several approaches such as the use of a secure hash function together with encryption or use of a message authentication code (MAC), have been adopted to detect data manipulation such as insertion, deletion, and substitution.

Authentication ensures the identification of either the data (*Data Origin Authentication*) or the entity (*Entity Authentication*). *Data origin authentication* implicitly provides data integrity since the unauthorised alteration of the data implies that the origin of the data is changed, as the origin of data can only be guaranteed if the data integrity has not been compromised in any way. The use of a one-way hash function together with encryption or use of a message authentication code (MAC) can help to achieve data origin authentication. *Entity authentication* is a communication process by which a principal establishes a live correspondence with a second principal whose identity should be that which is sought by the first principal. In cryptographic protocols, both entity authentication and data origin authentication are essential to establish the key.

Cryptographic protocols are designed to provide one or more of these security services between communicating agents in a hostile environment. To achieve confidentiality of data in a session established by some entity A, with another intended entity B, one may use a cryptographic primitive, called *symmetric key encryption*. This cryptographic algorithm produces a ciphertext message, c, when given some plaintext message, m. A then sends B the ciphertext c over the insecure communication channel. Only B who has a pre-established secret information (with A), known as a *shared key*, can decrypt c to obtain m, achieving the notion of data confidentiality.

The shared key can be a long-term key associated with some identities, a symmetric encryption key shared between two entities, or a session key. In a real world setting, it is normal to assume that a host can establish several concurrent sessions with many different parties. Therefore, the session key has to be fresh and unique for each session as sessions are specific to both the communicating parties.

The above security services are usually meaningful when guaranteed during a complete session of closely related interactions over a communication channel and in many cases, open and insecure communication channels. In most of these cases, there is a need for some temporary keys. For example, an encryption key for a shared-key encryption scheme in the above-mentioned scenario.

The advantages of using temporary (session) keys relative to using long-term keys directly are four-fold:

1) to limit the amount of cryptographic material available to cryptanalytic attacks;
2) to limit the exposure of messages when keys are lost,
3) to create independence between different and unrelated sessions, and
4) to achieve efficiency, e.g., if long-term keys are based on asymmetric cryptography, using session keys based on (faster) symmetric cryptography can bring a considerable gain in efficiency.

The establishment of session keys often involves interactive cryptographic protocols or also known as authentication and/or key establishment protocols. Such protocols are increasingly being considered as the *sine qua non* of many diverse secure electronic communications and electronic commerce applications.

Although technology advances have brought us many conveniences and benefits, they have also resulted in the erosion of many assumptions about the design of cryptographic protocols, which began in the 1970s. As a result, the environment for cryptographic protocols has changed drastically over the years. One thing that does not change with time is that the design of cryptographic protocols is still notoriously hard. Frequently, errors were found in many such protocols years after they were published (Abdalla *et al.*, 2006; Bao, 2003; Bao, 2004; Basin *et al.*, 2003; Choo, 2006b; Choo, 2006a; Lowe, 1996; Nam *et al.*, 2004; Pereira and Quisquater, 2003; Shoup, 2001; Wan and Wang, 2004; Wong and Chan, 2001). Hence, a high level of assurance is needed in the correctness of such protocols.

The study of cryptographic protocols has led to the dichotomization of cryptographic protocol analysis techniques between the computational complexity approach (Bellare *et al.*, 2000; Bellare and Rogaway, 1993; Bellare and Rogaway, 1995; Bellare *et al.*, 1998; Canetti and Krawczyk, 2001; Shoup, 2001) and the computer security approach (Fidge, 2001; Meadows, 2001; Meadows, 2003).

Computer Security Approachy. Emphasis in the computer security approach is placed on automated machine specification and analysis – using formal methods. Emphasis in the computer security approach is placed on automated machine specification and analysis. Researchers have attempted to verify, prove, and/or design cryptographic protocols with automated theorem provers (Barthe *et al.*, 2004; Lynch, 1999; Paulson, 1997), model checkers (Backes, 2004a; Backes, 2004b; Clarke *et al.*, 2000), logic-based approaches (including belief logic) (Aiello and Massacci, 2001; Gupta and Shmatikov, 2005), and other tools (including specific cryptographic protocol programming languages) (Allamigeon and Blanchet, 2005; Bodei *et al.*, 2003; Buccafurri *et al.*, 1999; Perrig and Song, 2000a; Perrig and Song, 2000b). The main goal is to relieve humans of the tedious and error prone parts of the mathematical proofs.

The Dolev and Yao (1983) adversarial model is the de-facto model used in formal specifications, where cryptographic operations are often used in a "black box" fashion ignoring the various cryptographic properties. This resulted in possible loss of partial information. For the foreseeable future, this approach requires abstractions of cryptographic primitives because the tools cannot handle the cryptographic details. However, all the classical abstractions were made ad-hoc.

One of the main obstacles in this automated approach is the undecidability and intractability problems since the adversary can have an exponentially large set of possible actions (or combinations) which result in a state explosion (Cervesato *et al.*, 1999). Furthermore, protocols proven secure in such a manner could possibly be flawed (i.e., giving a false positive result – analogous to a Type II error in hypothesis testing) (Backes and Jacobi, 2003). From a real world practicality perspective, it is debatable whether proofs of security in this manner carry significant weight in the real world, due to their idealistic model. However, the computer security approach should be credited for proving insecurities in protocols (i.e., finding both known and previously unknown flaws in protocols) (Allamigeon and Blanchet, 2005; Basin *et al.*, 2003).

Computational Complexity Approachy. Emphasis in the computational complexity approach is placed on a proven reduction from the problem of breaking the protocols to another problem believed to be hard. Application of the computational complexity approach to protocol analysis was initiated by Bellare and Rogaway (1993), with a proof for two-party entity authentication and key exchange protocols (Bellare and Rogaway, 1993). They formally defined a model of adversary capabilities with an associated definition of security. Since then, there have been several extensions to the Bellare and Rogaway (1993) proof model, such as the Bellare and Rogaway (1995) key establishment model (Bellare and Rogaway, 1995), Bellare, Pointcheval and Rogaway (2000) password-based mutual authentication and key establishment model (Bellare *et al.*, 2000), the Bresson, Chevassut and Pointcheval (2001) group authenticated key establishment model (Bresson *et al.*, 2001), and the most recent Abdalla, Fouque and Pointcheval (2005) password-based authenticated key establishment model (Abdalla *et al.*, 2005).

However, it is often difficult to obtain correct computational proofs of security. As Koblitz and Menezes (2004) had pointed out, computational proofs usually entail lengthy and complicated mathematical proofs, which are daunting to most readers. Difficulties in obtaining correct computational proofs of protocol security are evidenced by the breaking of provably-secure protocols after they were published (Choo, 2006a; Nam *et al.*, 2004; Wan and Wang, 2004; Wong and Chan, 2001). Despite these setbacks, proofs are invaluable for arguing about security and certainly are one very important tool in getting protocols right. We refer the reader to the protocol lounge for a list of published provably-secure protocols (Choo, 2004).

Case Study. In this work, we advocate the importance of proofs of protocol security, and by identifying some situations where errors in proofs arise, we hope that similar structural mistakes can be avoided in future proofs. As a case study, we revisit the password-based group key exchange protocol due to (Lee *et al.*, 2004), which carries a claimed proof of security in the Bresson *et al.* model under the intractability of the Decisional Diffie–Hellman problem (DDH) and Computational Diffie–Hellman (CDH) problem. We reveal a previously unpublished flaw in the protocol and its proof, whereby we demonstrate that the protocol violate the definition of security in the Bresson *et al.* model. We also revisit the authenticated key establishment protocol (i.e., protocol 4) due to (Song and Kim, 2000), and revealed a previously unpublished flaw in the protocol.

Organization of Paper. The remainder of this paper is structured as follows: Section 2 briefly explains the Bresson *et al.* model, which is an extension of the Bellare–Rogaway models in the group setting. Section 3 describes the Lee–Hwang–Lee password-based group key exchange protocol. Previously unpublished attack on the protocol is demonstrated. We conclude this section by proposing fix to the protocol. Fixed protocol is not proven secure, and is presented mainly to provide a better insight into the proof failure. Section 4 describes the Song–Kim protocol 4, and its previously unpublished attack. Section 5 presents the conclusions.

2. Informal Overview of the Bresson *et al.* Model

In the Bresson *et al.* model, the adversary $\mathcal{A}$ is defined to be a probabilistic machine that is in control of all communications between parties by interacting with two sets, $\Pi^i_{U_1,U_2}$ and $\Psi^j_{U_1,U_2}$ of oracles ($\Pi^i_{U_1,U_2}$ is defined to be the ith instantiation of a principal U_1 in a specific protocol run and U_2 is the principal with whom U_1 wishes to establish a secret key and $\Psi^j_{U_1,U_2}$ is defined to be the jth instantiation of the server in a specific protocol run establishing a shared secret key between U_1 and U_2). The predefined oracle queries are shown in Table 1.

2.1. *Definition of Partnership*

Partnership in the model is defined based on the notion of session identifiers (SIDs) where SIDs are defined to be the concatenation of messages exchanged during the protocol run.

Table 1

Informal description of the oracle queries

$\mathsf{Send}(U_1, U_2, i, m)$ query computes a response according to the protocol specification and decision on whether to accept or reject yet, and returns them to $\mathcal{A}$.
The client oracle, $\Pi^i_{U_1,U_2}$, upon receiving a $\mathsf{Reveal}(U_1, U_2, i)$ query, and if it has accepted and holds some session key, will send this session key back to $\mathcal{A}$.
$\mathsf{Corrupt}(U_1, K_E)$ query allows $\mathcal{A}$ to corrupt the principal U_1 at will, and thereby learn the complete internal state of the corrupted principal. The corrupt query also gives $\mathcal{A}$ the ability to overwrite the long-lived key of the corrupted principal with any value of her choice (i.e., K_E).
$\mathsf{Test}(U_1, U_2, i)$ query is the only oracle query that does not correspond to any of $\mathcal{A}$'s abilities. If $\Pi^i_{U_1,U_2}$ has accepted with some session key and is being asked a $\mathsf{Test}(U_1, U_2, i)$ query, then depending on a randomly chosen bit b, $\mathcal{A}$ is given either the actual session key or a session key drawn randomly from the session key distribution.

In this model, an oracle who has accepted will hold the associated session key, a SID and a partner identifier (PID). Definition 1 describes partnership in the model.

DEFINITION 1 (Definition of Partnership). Two oracles, $\Pi^i_{A,B}$ and $\Pi^j_{B,A}$, are partners if, and only if, both oracles have accepted the same session key with the same SID, have agreed on the same set of principals (i.e., the initiator and the responder of the protocol).

2.2. *Definition of Freshness*

Freshness is used to identify the session keys about which $\mathcal{A}$ ought not to know anything because $\mathcal{A}$ has not revealed any oracles that have accepted the key and has not corrupted any principals knowing the key. Definition 2 describes freshness, which depends on the notion of partnership. Note that we do not consider the notion of forward secrecy in this paper, otherwise, the definition of freshness would be slightly different.

DEFINITION 2 (Definition of Freshness). Oracle $\Pi^i_{A,B}$ is fresh (or holds a fresh session key) at the end of execution, if, and only if,

1) $\Pi^i_{A,B}$ has accepted with or without a partner oracle $\Pi^j_{B,A}$;
2) both $\Pi^i_{A,B}$ and $\Pi^j_{B,A}$ oracles have not been sent a Reveal query, and
3) A and B have not been sent a $\mathsf{Corrupt}$ query.

2.3. *Definition of Security*

Security in the Bellare–Rogaway and the Canetti–Krawczyk models is defined using the game $\mathcal{G}$, played between a malicious adversary $\mathcal{A}$ and a collection of $\Pi^i_{U_x,U_y}$ oracles for players $U_x, U_y \in \{U_1, \ldots, U_{N_p}\}$ and instances $i \in \{1, \ldots, N_s\}$. The adversary $\mathcal{A}$ runs the game $\mathcal{G}$, whose setting is explained in Table 2.

Success of $\mathcal{A}$ in $\mathcal{G}$ is quantified in terms of $\mathcal{A}$'s advantage in distinguishing whether $\mathcal{A}$ receives the real key or a random value. $\mathcal{A}$ wins if, after asking a $\mathsf{Test}(U_1, U_2, i)$ query, where $\Pi^i_{U_1,U_2}$ is fresh and has accepted, $\mathcal{A}$'s guess bit b' equals the bit b selected during the $\mathsf{Test}(U_1, U_2, i)$ query. Let the advantage function of $\mathcal{A}$ be denoted by $\mathsf{Adv}^{\mathcal{A}}(\mathsf{k})$, where

$$\mathsf{Adv}^{\mathcal{A}}(\mathsf{k}) = 2 \times \Pr[\mathsf{b} = \mathsf{b}'] - 1.$$

Table 2

Setting of game $\mathcal{G}$

Stage 1:	$\mathcal{A}$ is able to send any oracle queries at will.
Stage 2:	At some point during $\mathcal{G}$, $\mathcal{A}$ will choose a fresh session on which to be tested and send a Test query to the fresh oracle associated with the test session. Depending on the randomly chosen bit b, $\mathcal{A}$ is given either the actual session key or a session key drawn randomly from the session key distribution.
Stage 3:	$\mathcal{A}$ continues making any oracle queries at will but cannot make $\mathsf{Corrupt}$ and/or Reveal that trivially expose the test session key.
Stage 4:	Eventually, $\mathcal{A}$ terminates the game simulation and outputs a bit b', which is its guess of the value of b.

Definition 3 describes the definition of security.

DEFINITION 3 (Definition of Security). A protocol is secure in the model if both the following requirements are satisfied:

1. When the protocol is run in the absence of a malicious adversary, all partner oracles accept and hold the same session key.
2. For all probabilistic, polynomial-time (PPT) adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathcal{A}}(\mathsf{k})$ is negligible.

Now that the model has been defined, we can define the key replicating attack. The key replicating attack, first introduced by Krawczyk (Krawczyk, 2005), will be referred to in this paper and is described in Definition 4.

DEFINITION 4 (Key Replicating Attack (Krawczyk, 2005)). A key replicating attack is defined to be an attack whereby the adversary, $\mathcal{A}$, succeeds in forcing the establishment of a session, S_1, (other than the Test session or its matching session) that has the same key as the Test session. In this case, $\mathcal{A}$ can distinguish whether the Test-session key is real or random by asking a Reveal query to the oracle associated with S_1.

3. Lee *et al.* (2004) Password-Based Group Key Exchange Protocol

The notations used in the protocol is presented in Table 3.

Fig. 1 describes the password-based group key exchange protocol due to Lee *et al.* (2004). In the protocol, members of the same group, $\mathcal{U} = \{U_1, U_2, U_3, \ldots, U_4\}$ are assumed to be honest (i.e., The adversary, $\mathcal{A}$, is assumed not to be a member of $\mathcal{U}$) and sharing a secret password, pwd.

To establish a group session key, each user U_i selects a random number $x_i \in_R \mathbb{Z}_q^*$ and sends a message $ID_1||\mathcal{E}(g^{x_i})_{pwd}$ to all group users participating in the execution of the protocol. Upon receiving $ID_j||\mathcal{E}(g^{x_j})_{pwd}$ (where $i \neq j$), each user U_i computes $w_1 =$

Table 3

Summary of notations

pwd	denotes some secret password share between A and B
$\mathcal{E}(\cdot)_{pwd}$	denotes the encryption of some message under the password, pwd
$\mathcal{H}(\cdot)$, $h(\cdot)$	denotes the hashes of some message, where $\mathcal{H}$ and h are independent hash functions
ID_U	denotes the identity of some entity, U
sk_U	denotes the secret session key of some entity, U
SID_U	denotes the session identifier (SID) of some entity, U
PID_U	denotes the partner identifier (PID) of some entity, U

U_1	$\dots$	U_n
$x_1 \in_R \mathbb{Z}_q^*$		$x_n \in_R \mathbb{Z}_q^*$
Broadcast $ID_1\|\|\mathcal{E}(g^{x_1})_{pwd}$		**Broadcast** $ID_n\|\|\mathcal{E}(g^{x_n})_{pwd}$
$w_1 = h(g^{x_n x_1}) \oplus h(g^{x_1 x_2})$		$w_n = h(g^{x_{n-1} x_n}) \oplus h(g^{x_n x_1})$
Broadcast $ID_1\|\|w_1$		**Broadcast** $ID_n\|\|w_n$

Compute $sk_1 = \mathcal{H}(h(g^{x_1 x_2})||h(g^{x_2 x_3})||\dots||h(g^{x_{n-1} x_n})) = \dots = sk_n$

Fig. 1. Lee *et al.* (2004) password-based group key exchange protocol.

$h(g^{x_{i-1}x_i}) \oplus h(g^{x_i x_{i+1}})$ and broadcasts a message $ID_i||w_i$. Upon receiving $ID_j||w_j$ (where $i \neq j$), each user U_i computes:

$$h^{(g^{x_{j-1}x_j})} = w_j \oplus h^{(g^{x_j x_{j+1}})} = h^{(g^{x_{j-1}x_j})} \oplus h^{(g^{x_j x_{j+1}})} \oplus h^{(g^{x_j x_{j+1}})},$$
$$sk_i = \mathcal{H}\big(h^{(g^{x_1 x_2})}||\dots||h^{(g^{x_{n-1}x_n})}||h^{(g^{x_n x_1})}\big).$$

3.1. *New Attack*

Fig. 2 describes the scenario where U_1 wishes to establish a session key with only U_2 and U_3, in the presence of a malicious adversary, $\mathcal{A}$.

At the end of the protocol execution, U_2 and U_3 think that the session key is being shared with U_4, when in fact the session key is being shared with U_1. In other words, $PID_{U_1} = \{U_2, U_3\}$, $PID_{U_2} = \{U_3, U_4\}$, and $PID_{U_3} = \{U_2, U_4\}$. This is also known

U_1	$\dots$	U_n
$x_1 \in_R \mathbb{Z}_q^*$		$x_n \in_R \mathbb{Z}_q^*$
Broadcast $ID_1\|\|\mathcal{E}(g^{x_1})_{pwd}$		**Broadcast** $ID_n\|\|\mathcal{E}(g^{x_n})_{pwd}$

$\mathcal{A}$ intercepts $ID_1||\mathcal{E}(g^{x_1})_{pwd}$
$\mathcal{A}$ impersonates U_4 and broadcasts $ID_4||\mathcal{E}(g^{x_1})_{pwd}$

$w_1 = h(g^{x_n} g^{x_1}) \oplus h(g^{x_1} g^{x_2})$		$w_n = h(g^{x_{n-1}} g^{x_n}) \oplus h(g^{x_n} g^{x_1})$
Broadcast $ID_1\|\|w_1$		**Broadcast** $ID_n\|\|w_n$

$\mathcal{A}$ intercepts $ID_1||w_1$
$\mathcal{A}$ impersonates U_4 and broadcasts $ID_4||w_1$

Compute $sk_1 = \mathcal{H}(h(g^{x_1} g^{x_2})||h(g^{x_2} g^{x_3})||\dots||h(g^{x_{n-1}} g^{x_n})) = \dots = sk_n$

Fig. 2. Execution of Lee *et al.* (2004) password-based group key exchange protocol in the presence of a malicious adversary.

as an unknown key share attack. Unknown key share attack was first discussed by Diffie, van Oorschot and Wiener in 1992. $\mathcal{A}$ needs not obtain the session key and still profits from this attack. Consider the scenario whereby A will deliver some information of value (such as e-cash) to B. Since B believes the session key is shared with $\mathcal{A}$, $\mathcal{A}$ can claim that this credit deposit as his (Boyd and Mathuria, 2003; Chapter5.1.2).

In the context of the Bresson *et al.* model, $\mathcal{A}$ can trivially expose a fresh session key by revealing either U_2 or U_3 since neither U_2 nor U_3 are partners of U_1. Hence, the Lee–Hwang–Lee password-based group key exchange protocol shown in Fig. 1 is not secure since the adversary $\mathcal{A}$ is able to obtain the fresh session key of the initiator U_1 by revealing non-partner oracles of U_1 (i.e., U_2 or U_3), in violation of the key establishment goal. This is also known as a key replicating attack described in Definition 4.

3.2. *Preventing the Attack*

The countermeasures are well studied and we may adopt the same approach by Choo, Boyd and Hitchcock (2005) who suggest that

– Including the identities of the participants and their roles in the key derivation function provides resilience against unknown key share attacks (Boyd and Mathuria, 2003; Chapter 5.1.2) and reflection attacks (Krawczyk, 2003), and

– Including the transcripts in the key derivation function provides freshness and data origin authentication.

Hence, a possible fix for the protocol is to include the sender's identity in each encryption and also the session identifier, sid, in the key derivation function, as shown in Fig. 3.

We use the same construct for sid (i.e., the concatentation of all messages received) as used by Lee, Hwang and Lee. In the improved protocol, the adversary $\mathcal{A}$ will not be able to falsify ownership of the encrypted message $\mathcal{E}(U_1||g^{x_1})_{pwd}$ since the identity of the sender is included in the encryption. Since the construct of the session key in the improved protocol comprises the associated sid, a different sid will imply a different session key. Hence, the attack shown in Fig. 2 will no longer be valid against this improved protocol.

U_1	$\ldots$	U_n
$x_1 \in_R \mathbb{Z}_q^*$		$x_n \in_R \mathbb{Z}_q^*$
Broadcast $ID_1||\mathcal{E}(ID_1||g^{x_1})_{pwd}$		**Broadcast** $ID_n||\mathcal{E}(ID_n||g^{x_n})_{pwd}$
$w_1 = h(ID_1||g^{x_n x_1}) \oplus h(ID_1||g^{x_1 x_2})$		$w_n = h(ID_n||g^{x_{n-1} x_n}) \oplus h(ID_n||g^{x_n x_1})$
Broadcast $ID_1||w_1$		**Broadcast** $ID_n||w_n$

$$sid_1 = \mathcal{E}(ID_1||g^{x_1})_{pwd}||\ldots||\mathcal{E}(ID_n||g^{x_n})_{pwd}||w_1||\ldots||w_n = sid_n$$

$$\text{Compute } sk_1 = \mathcal{H}(h(g^{x_1 x_2})||h(g^{x_2 x_3})||\ldots||h(g^{x_{n-1} x_n})||sid_1) = \ldots = sk_n$$

Fig. 3. An improved Lee *et al.* (2004) password-based group key exchange protocol.

3.3. *Flaws in Existing Proof*

The existing proof assumes that $\mathsf{Adv}^{\mathcal{A}}(\mathsf{k})$ is negligible. Consider the attack outlined in Fig. 2 where $\mathcal{A}$ is able to distinguish a real key or a random key by asking a Reveal query to a non-partner server oracle of A, and hence violate the key establishment goal with non-negligible probability. The DDH and CDH breaker $\mathcal{A}_{\mathcal{DDH}/\mathcal{CDH}}$ (which is constructed using $\mathcal{A}$) is unable to obtain a non-negligible probability of breaking the DDH and CDH problems, contradicting the underlying assumption in the proof. Consequently, the proof simulation fails (the result of Send and Reveal queries were not adequately considered in the simulation).

4. Song–Kim Key Agreement Protocol 4

Protocol AK (Song and Kim, 2000) in Fig. 4 involves two parties, A and B. The notation used in the protocol is as follows, $r_A \in_R \mathbb{Z}_n$ denote that r_A is randomly drawn from $\mathbb{Z}_n$, $(W_A = w_AP, w_A)$ and $(W_B = w_BP, w_B)$ denote the public/private key pair of A and B respectively, and P denotes the base point in the elliptic curve.

The security goals of this protocol are entity authentication and key establishment. At the end of the protocol execution shown in Fig. 4, both A and B accept the same secret key

$$SK_A = cr_AW_B + c(w_A + r_A)R_B = c(r_Aw_B + r_Bw_A + r_Ar_B)P,$$
$$SK_B = cr_BW_A + c(w_B + r_B)R_A = SK_A.$$

Fig. 5 presents the execution of Song–Kim key agreement protocol 4 in the presence of a malicious adversary, $\mathcal{A}$.

The attack sequence shown in Fig. 5 is as follows: the protocol starts when A wants to establish a session with B and sends $R_A = r_AP$. The adversary $\mathcal{A}$ intercepts and deletes this message meant for B, and instead impersonate B and sends this message to A. A, upon receiving this message thinks that B wants to establish a connection in another session run (S2), will respond as per the protocol specification with $R_{A2} = r_{A2}P$. Again, $\mathcal{A}$ intercepts and deletes $R_{A2} = r_{A2}P$ meant for B, and instead impersonate B and sends $R_{A2} = r_{A2}P$ back to A. In both sessions S1 and S2, A has accepted session keys $SK_{A(S1)}$ and $SK_{A(S2)}$, where $SK_{A(S1)} = cr_AW_B + c(w_A + r_A)R_{A2} = SK_{A(S2)}$.

A		B
$r_A \in_R \mathbb{Z}_n$		
$R_A = r_AP$	$\xrightarrow{R_A}$	$r_B \in_R \mathbb{Z}_n$
	$\xleftarrow{R_B}$	$R_B = r_BP$
$SK_A = cr_AW_B + c(w_A + r_A)R_B$		$SK_B = cr_BW_A + c(w_B + r_B)R_A$

Fig. 4. Song–Kim key agreement protocol 4.

$1(S1)$.	$\mathsf{A} \longrightarrow \mathsf{B}:$	$R_A = r_A P$
		$\mathcal{A}$ intercepts message $R_A = r_A P$ meant for B.
$1(S2)$.	$\mathsf{A}_\mathsf{B} \longrightarrow \mathsf{A}:$	$R_A = r_A P$
$2(S2)$.	$\mathsf{A} \longrightarrow \mathsf{B}:$	$R_{A2} = r_{A2} P$
		$\mathcal{A}$ intercepts message $R_{A2} = r_{A2} P$ meant for B.
$1(S1)$.	$\mathsf{A}_\mathsf{B} \longrightarrow \mathsf{A}:$	$R_{A2} = r_{A2} P$
$3(S2)$.	$\mathcal{A} \longrightarrow \mathsf{A}:$	Reveal
$4(S2)$.	$\mathsf{A} \longrightarrow \mathsf{B}:$	$SK_{A(S2)}$

Fig. 5. Reflection attack on Song–Kim key agreement protocol 4.

In session S1, A is the initiator (with B being the perceived responder) and in session S2, A is the responder (with B being the perceived initiator). However, B is not aware of both sessions. However, according to Definition 2.1, A has no partner. Hence, $\mathcal{A}$ is able to reveal the session key accepted by A in Session 2 and obtain the session key in Session 1.

Hence, the Song–Kim AK protocol shown in Fig. 4 is not a secure authenticated key establishment protocol, since the adversary $\mathcal{A}$ is able to violate both the entity authentication and key establishment goals of this protocol as shown in Fig. 5.

This attack supports the observation by Blake–Wilson, Johnson and Menezes (1997) that two-flow authenticated key establishment protocols that do not contain asymmetry in the formation of the session key will not meet the security requirements in the Bellare–Rogaway model (Blake–Wilson *et al.*, 1997).

5. Conclusion

Through a detailed study of the password-based group key exchange protocol due to Lee *et al.* (2004) we have concluded that specifying correct computational complexity proofs for protocols remains a hard problem. However, we have identified an area where protocol proofs are likely to fail, namely Send and Reveal queries not adequately considered in the proof simulations. We may speculate that the flaws in protocols with claimed proofs of security could have been discovered by the protocol designers if complete proof specifications had been constructed.

Through the study of the authenticated key establishment protocol due to Song and Kim (2000), we have concluded that proofs are invaluable for arguing about security and certainly are one very important tool in getting protocols right. Without proofs of security, protocol implementers cannot be assured about the security properties of protocols. Flaws in protocols discovered after they were published or implemented certainly will have a damaging effect on the trustworthiness and the credibility of key establishment protocols in the real world. As a result of this work, we would recommend that protocol designers provide proofs of security for their protocols, in order to assure protocol implementers about the security properties of protocols.

Acknowledgments

The author would like to thank the anonymous reviewers for providing valuable comments concerning this paper. Despite their invaluable assistance any errors remaining in this paper are solely attributed to the author.

References

Abdalla, M., E. Bresson, O. Chevassut and D. Pointcheval (2006). Password-based group key exchange in a constant number of rounds. In M. Yung *et al.* (Eds.), *PKC 2006*, vol. 3958/2006 of *LNCS*, Springer-Verlag. pp. 427–442.

Abdalla, M., P.-A. Fouque and D. Pointcheval (2005). Password-based authenticated key exchange in the three-party setting. In S. Vaudenay (Ed.), *PKC 2005*, vol. 3386/2005 of *LNCS*, Springer-Verlag. pp. 65–84.

Aiello, L.C., and F. Massacci (2001). Verifying security protocols as planning in logic programming. *ACM Transactions on Computational Logic* (*Special Issue Devoted to Robert A. Kowalski*), **2**(4), 542–580.

Allamigeon, X., and B. Blanchet (2005). Reconstruction of attacks against cryptographic protocols. In *CSFW 2005*. IEEE Computer Society Press. pp. 140–154.

Backes, M. (2004a). A cryptographically sound dolev-yao style security proof of the Needham–Schroeder–Lowe public-key protocol. *IEEE Journal on Selected Areas in Communications*, **22**(10), 2075–2086.

Backes, M. (2004b). A cryptographically sound dolev-yao style security proof of the Otway–Rees protocol. In P. Samarati and D. Gollmann (Eds.), *ESORICS 2004*, vol. 3193/2004 of *LNCS*. Springer-Verlag. pp. 89–108.

Backes, M., and Ch. Jacobi (2003). Cryptographically sound and machine-assisted verification of security protocols. In H. Alt and M. Habib (Eds.), *STACS 2003*, vol. 2607/2003 of *LNCS*. Springer-Verlag. pp. 310–329.

Bao, F. (2003). Security analysis of a password authenticated key exchange protocol. In C. Boyd and W. Mao (Eds.), *6th Information Security Conference – ISC 2003*, vol. 2851/2003 of *LNCS*. Springer-Verlag. pp. 208–217.

Bao, F. (2004). Colluding attacks to a payment protocol and two signature exchange schemes. In P.J. Lee (Ed.), *ASIACRYPT 2004*, vol. 3329/2004 of *LNCS*. Springer-Verlag. pp. 417–429.

Barthe, G., J. Cederquist and S. Tarento (2004). A machine-checked formalization of the generic model and the random oracle model. In D.A. Basin and M. Rusinowitch (Eds.), *IJCAR 2004*, vol. 3097/2005 of *Lecture Notes in Computer Science*. Springer-Verlag. pp. 385–399.

Basin, D.A., S. Mödersheim and L. Viganó (2003). An on-the-fly model-checker for security protocol analysis. In E. Snekkenes and D. Gollmann (Eds.), *ESORICS2003*, vol. 2808/2003 of *LNCS*. Springer-Verlag. pp. 253–270.

Bellare, M., R. Canetti and H. Krawczyk (1998). A modular approach to the design and analysis of authentication and key exchange protocols. In J. Vitter (Ed.), *ACM STOC 1998*. ACM Press. pp. 419–428.

Bellare, M., D. Pointcheval and P. Rogaway (2000). Authenticated key exchange secure against dictionary attacks. In B. Preneel (Ed.), *EUROCRYPT 2000*, vol. 1807/2000 of *LNCS*. Springer-Verlag. pp. 139–155.

Bellare, M., and P. Rogaway (1993). Entity authentication and key distribution. In D.R. Stinson (Ed.), *CRYPTO 1993*, vol. 773/1993 of *LNCS*. Springer-Verlag. pp. 110–125.

Bellare, M., and P. Rogaway (1995). Provably secure session key distribution: the three party case. In F.T. Leighton and A. Borodin (Eds.), *ACM STOC 1995*. ACM Press. pp. 57–66.

Blake–Wilson, S., D. Johnson and A. Menezes (1997). Key agreement protocols and their security analysis. In M. Darnell (Ed.), *IMA Cryptography and Coding 1997*, vol. 1335/1997 of *LNCS*. Springer-Verlag. pp. 30–45.

Bodei, C., M. Buchholtz, P. Degano, F. Nielson and H.R. Nielson (2003). Automatic validation of protocol narration. In R. Focardi (Ed.), *CSFW 2003*. IEEE Computer Society Press. pp. 126–140.

Boyd, C., and A. Mathuria (2003). *Protocols for Authentication and Key Establishment*. Springer-Verlag.

Bresson, E., O. Chevassut and D. Pointcheval (2001). Provably authenticated group Diffie–Hellman key exchange – the dynamic case. In C. Boyd (Ed.), *ASIACRYPT 2001*, vol. 2248/2001 of *LNCS*. Springer-Verlag. pp. 209–223.

Buccafurri, F., T. Eiter, G. Gottlob and N. Leone (1999). Enhancing model checking in verification by AI techniques. *Artificial Intelligence*, **112**(1–2), 57–104.

Canetti, R., and H. Krawczyk (2001). Analysis of key-exchange protocols and their use for building secure channels. In B. Pfitzmann (Ed.), *EUROCRYPT 2001*, vol. 2045/2001 of *LNCS*. Springer-Verlag. pp. 453–474. (Extended version available from `http://eprint.iacr.org/2001/040/`).

Cervesato, I., N. Durgin, P.D. Lincoln, J.C. Mitchell and A. Scedrov (1999). A meta-notation for protocol analysis. In P. Syverson (Ed.), *CSFW 1999*. IEEE Computer Society Press. pp. 55–71.

Choo, K.-K.R. (2006a). *Key Establishment: Proofs and Refutations*. Ph.D. Thesis, Information Security Institute, Queensland University of Technology.

Choo, K.-K.R. (2006b). Refuting security proofs for tripartite key exchange with model checker in planning problem setting. In *CSFW 2006*. IEEE Computer Society Press. pp. 297–308.

Choo, K.-K.R. (2004). The provably-secure key establishment and mutual authentication protocols lounge. `http://sky.fit.qut.edu.au/ choo/lounge.html`.

Choo, K.-K.R., C. Boyd and Y. Hitchcock (2005a). On session key construction in provably secure protocols. In E. Dawson and S. Vaudenay (Eds.), *MYCRYPT 2005*, vol. 3715/2005 of *LNCS*. Springer-Verlag. pp. 116–131.

Choo, K.-K.R., C. Boyd and Y. Hitchcock (2005b). Errors in computational complexity proofs for protocols. In B. Roy (Ed.), *ASIACRYPT 2005*, vol. 3788/2005 of *LNCS*. Springer-Verlag. pp. 624–643.

Clarke, E.M., S. Jha and W. Marrero (2000). Verifying security protocols with brutus. *ACM Transactions on Software Engineering and Methodology*, **9**(4), 443–487.

Diffie, W., P.C. van Oorschot and M.J. Wiener (1992). Authentication and authenticated key exchange. *Journal of Designs, Codes and Cryptography*, **2**, 107–125.

Dolev, D., and A.C. Yao (1983). On the security of public key protocols. *IEEE Transaction of Information Technology*, **29**(2), 198–208.

Fidge, C.J. (2001). *A Survey of Verification Techniques for Security Protocols*. Technical report 01-22, Software Verification Research Centre, The University of Queensland, Brisbane.

Gupta, P., and V. Shmatikov (2005). Towards computationally sound symbolic analysis of key exchange protocols. In V. Atluri *et al.* (Eds.), *FMSE 2005*, ACM Press. pp. 23–32. (Full version available from `http://eprint.iacr.org/2005/171`)

Koblitz, N., and A. Menezes (2004). *Another Look at "Provable Security"*. Technical report CORR 2004-20, Centre for Applied Cryptographic Research, University of Waterloo, Canada. (Also available from `http://eprint.iacr.org/2004/152/`)

Krawczyk, H. (2003). SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie–Hellman and its use in the IKE-protocols. In D. Boneh (Ed.), *CRYPTO 2003*, vol. 2729/2003 of *LNCS*. Springer-Verlag. pp. 400–425.

Krawczyk, H. (2005). HMQV: a high-performance secure Diffie–Hellman protocol. In V. Shoup (Ed.), *CRYPTO 2005*, vol. 3621/2005 of *LNCS*. Springer-Verlag. pp. 546–566. (Extended version available from `http://eprint.iacr.org/2005/176/`)

Lee, S.M., J.Y. Hwang and D.H. Lee (2004). Efficient password-based group key exchange. In S. Katsikas, J. Lopez and G. Pernul (Eds.), *Trust and Privacy in Digital Business – TrustBus 2004*, vol. 3184/2004 of *LNCS*. Springer-Verlag. pp. 191–199.

Lowe, G. (1996). Some new attacks upon security protocols. In *CSFW 1996*. IEEE Computer Society Press. pp. 162–169.

Lynch, N.A. (1999). I/O automaton models and proofs for shared-key communication systems. In P. Syverson (Ed.), *CSFW 1999*. IEEE Computer Society Press. pp. 14–29.

Meadows, C. (2001). Open issues in formal methods for cryptographic protocol analysis. In *DARPA Information Survivability Conference and Exposition*, vol. 2052. IEEE Computer Society Press. pp. 237–250.

Meadows, C. (2003). Formal methods for cryptographic protocol analysis: emerging issues and trends. *IEEE Journal on Selected Area in Communications*, **21**(1), 44–54.

Menezes, A.J., P.C. van Oorschot and S.A. Vanstone (1997). *Handbook of Applied Cryptography*. The CRC Press Series On Discrete Mathematics And Its Applications. CRC Press.

Nam, J., S. Kim and D. Won (2004). *Attacks on Bresson-Chevassut-Essiari-Pointcheval's Group Key Agreement Scheme*. Cryptology ePrint Archive, Report 2004/251. `http://eprint.iacr.org/2004/251/`.

Paulson, L.C. (1997). Proving properties of security protocols by induction. In *CSFW 1997*. IEEE Computer Society Press. pp. 70–83.

Pereira, O., and J.-J. Quisquater (2003). Some attacks upon authenticated group key agreement protocols. *Journal of Computer Security*, **11**, 555–580.

Perrig, A., and D. Song (2000a). A first step towards the automatic generation of security protocols. In *NDSS 2000*. Internet Society Press. pp. 73–83.

Perrig, A., and D. Song (2000b). Looking for diamonds in the desert: extending automatic protocol generation to three-party authentication and key agreement protocols. In *CSFW 2000*. IEEE Computer Society Press.

Shoup, V. (2001). OAEP reconsidered. In J. Kilian (Ed.), *CRYPTO 2001*, vol. 2139/2001 of *LNCS*. Springer-Verlag. pp. 239–259.

Song, B., and K. Kim (2000). Two-pass authenticated key agreement protocol with key confirmation. In B.K. Roy and E. Okamoto (Eds.), *INDOCRYPT 2000*, vol. 1977/2000 of *LNCS*. Springer-Verlag. pp. 237–249.

Wan, Z., and S. Wang (2004). Cryptanalysis of two password-authenticated key exchange protocols. In H. Wang, J. Pieprzyk and V. Varadharajan (Eds.), *ACISP 2004*, vol. 3108/2004 of *LNCS*. Springer-Verlag.

Wong, D.S., and A.H. Chan (2001). Efficient and mutually authenticated key exchange for low power computing devices. In C. Boyd (Ed.), *ASIACRYPT 2001*, vol. 2248/2001 of *LNCS*. Springer-Verlag. pp. 172–289.

K.-K.R. Choo is currently a high tech crime research analyst with the Australian Institute of Criminology Information. He received his BSc maths, BAppSci (hons) industrial & applied maths, master of information technology, graduate diploma in business administration, and PhD information security degrees in Dec 2000, Dec 2002, May 2002, Dec 2005, and Sep 2006 respectively. His research interests include key establishment protocols, provable security, and high crime.

Lee, Hwang ir Lee (2004) ir Song ir Kim (2000) raktų apsikeitimo / susitarimo protokolo saugumo analizė

Kim-Kwang Raymond CHOO

Mes pataisome slaptažodžiu paremto grupinio raktų apsikeitimo protokolo teorinius rezultatus, paskelbtus Lee, Hwang ir Lee (2004), kuriuose pateiktas saugumo įrodymas, panaudojant Bresson *et al.* modelį, remiantis algoritmiškai sunkiai sprendžiamomis Sprendimine Diffie–Helman'o (SpDH) problema ir skaičiuojamąja Diffie–Helman'o (SkDH) problema. Mes pateikiame anksčiau nepublikuotą šio protokolo trūkumą ir jo įrodymą, kuo parodome, kad protokolas prieštarauja modelyje nustatytam saugumo apibrėžimui. Tam, kad geriau suprasti protokolą ir įrodymo klaidas, mes pateikiame tam tikrą kitą protokolą. Mes manome, kad mūsų analizė leis išvengti panašių klaidų ateityje. Mes taip pat pataisome 4 Song'o ir Kim'o (2000) protokolą ir atskleidžiame anksčiau nepublikuotas jo klaidas (t.y. atspindžio ataką).