On the Security of Lee, Kim, Kim, & Oh Key Agreement Protocol

Kim-Kwang Raymond Choo

Information Security Institute, Queensland University of Technology
GPO Box 2434, Brisbane, QLD 4001, Australia (Email: k.choo@qut.edu.au)

(Received Oct. 3, 2005; revised and accepted Nov. 15 & 16, 2005)

Abstract

In ICCSA 2005, Lee, Kim, Kim, & Oh proposed a new (two-party) ID-based key agreement protocol, which they claimed to provide known key security resilience, forward secrecy, key compromise resilience, unknown key share resilience, and key control, however, without providing any security proofs. In this work, we demonstrate that their claims of known key security resilience and key control are flawed by revealing previously unpublished flaw in the two-party ID-based key agreement protocol. We may speculate that such (trivial) errors could have been found by protocol designers if proofs of security were to be constructed, and hope this work will encourage future protocol designers to provide proofs of security. We conclude with a countermeasure due to Choo, Boyd, & Hitchcock (2005).

Keywords: Identity-based protocol, key agreement protocol, proofs of security, provable 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 is 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 with being able to provide security guarantees becomes of paramount importance. Many initiatives have been proposed to address this concern; and cryptographic data encryption and authentication constitute the tools to address it. Typically security guarantees are provided by means of protocols that make use of security primitives such as encryption, digital signatures, and hashing.

Menezes, van Oorschot, & Vanstone [32] [Chapter 1] and Boyd & Anish Mathuria [11] [Chapter 1] identify the following possible different properties 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) gets hold of the encrypted data. In cryptographic protocols, confidentiality ensures that keys and other data are available to the authorised principals as intended and trusted third party server if applicable.

Data integrity guarantees the data has not been tampered with or modified. To achieve this notion, several approaches such as the use of a one-way 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. In cryptographic protocols, data integrity ensures that elements such as nonces and identity fields are protected.

Authentication ensures the identification, which can be of 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.

Non-repudiation ensures that entities cannot deny any previous commitments or actions. Non-repudiation provides data integrity and data origin authentication implying the origin of the data which in turn, implies the integrity of the data. Application of digital signature mechanisms helps to achieve this notion of non-repudiation. In cryptographic protocols, non-repudiation ensures that entities cannot deny any previous commitments or actions.

Cryptographic protocols are designed to provide one or more of these security guarantees between communicating agents in a hostile environment, e.g., to achieve confidentiality of data in a session established by some entity, A, with another intended entity, B, one may use a cryptographic algorithm, called *symmetric 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 session key*, that is fresh and unique for each session, can decrypt c to obtain m (i.e., achieving the notion of data confidentiality).

The above security properties 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 (e.g., 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 (since in a real world setting, it is normal to assume that a host can establish several concurrent sessions with many different parties. Sessions are specific to both the communicating parties), and

4) to achieve efficiency (e.g., if our long-term keys are based on asymmetric cryptography, using session keys based on (faster) symmetric cryptography can bring a considerable gain in efficiency).

The many flaws discovered in published protocols for key establishment and authentication over many years, have led to a dichotomy in cryptographic protocol analysis techniques between the computational complexity approach [1, 6, 8, 12, 33] and the computer security approach [19, 30, 31].

The computer security approach concentrates on designing tools to formally verify the security of cryptographic protocols. The computational complexity approach concentrates on designing provably secure protocols, which adopts a deductive reasoning process whereby the emphasis is placed on a proven reduction from the problem of breaking the protocol to another problem believed to be hard. Such an approach for key establishment protocols was made popular by Bellare & Rogaway. In fact, Bellare & Rogaway [6] provided the first formal definition for a model of adversarial capabilities with an associated definition of security (hereafter referred to as the BR93 model).

A complete (human-generated) mathematical proof with respect to cryptographic definitions provides a strong assurance that a protocol is behaving as desired. The difficulty of obtaining correct computational proofs of security, however, has been illustrated dramatically by the well-known problem with the OAEP mode for public key encryption [34]. Although OAEP was one of the most widely used and implemented algorithms, it was several years after the publication of the original proof that a problem was found (and subsequently fixed in the case of RSA). Problems with proofs of protocol security have occurred too. Furthermore, such security proofs usually entail lengthy and complicated mathematical proofs, which are daunting to most readers [24, 25]. The breaking of provably-secure protocols after they were published is evidence of the difficulty of obtaining correct computational proofs of protocol security. Despite these setbacks, proofs are invaluable for arguing about security and certainly are one very important tool in getting protocols right.

The BR93 model has been further revised several times. In 1995, Bellare and Rogaway analysed a three-party server-based key distribution (3PKD) protocol [7] using an extension to the BR93 model, which will be referred to as the BR95 model. A more recent revision to the model was proposed in 2000 by Bellare, Pointcheval and Rogaway [5], hereafter referred to as the BPR2000 model. Collectively, the BR93, BR95, and BPR2000 models will be referred to as the Bellare–Rogaway models.

In independent yet related work, Bellare, Canetti, & Krawczyk [4] built on the BR93 model and introduced a modular proof model. However, some drawbacks with this formulation were discovered and this modular proof model was subsequently modified by Canetti & Krawczyk [12], and will be referred to as the CK2001 model in this thesis. Although the trend towards such formal approaches has been gaining momentum in recent years, the number of protocols that possess a rigorous proof of security remains relatively small [14].

There exists many protocols whose purported security is based on heuristic security arguments. The main problem with protocols with only heuristic security arguments is that it lacks formal foundations, and suffers from the following problems:

- Since this approach does not account for all possible attacks, the security guarantees are limited and often insufficient.

- This approach does not provides a clear framework on a formal description for a "secure" protocol and what constitutes an "attack".

In the provable security approach for protocols, the description of protocols security and the goals provided by the protocols are formally defined (e.g., we will know whether a proposed attack is valid and what it means to be secure).

Case Study. In this work, we advocate the importance of proofs of protocol security and the proposal of any protocol should provide a rigorous proof of security (as we argue that protocols without any computational proofs of security leads one to question the level of trust in the correctness in such protocols). As a case study, we revisit a recent work of Lee, Kim, Kim, & Oh [27]. They proposed a novel two-party ID-based key agreement protocol, which they claimed to be provide known key security resilience, forward secrecy, key compromise resilience, unknown key share resilience, and key control, however, without providing any security proofs. They then extend the two-party protocol to tripartite setting.

We reveal previously unpublished flaw in the two-party ID-based key agreement protocol, and demonstrate that their claims of known key security resilience and key control are flawed. To better explain our attack, we recall:

- the BR93 model [6]. It is common knowledge that the Reveal query in the BR93 model captures the known key security property, and
- the key integrity property due to Janson & Tsudik [21] and the key replicating attack due to Krawczyk [26].

Organization of Paper. The remainder of this paper is structured as follows: Section 2 provides an informal overview of the BR93 model. In this section, we also recall the key integrity property first discussed by Janson & Tsudik [21]. Section 3 revisits the two-party ID-based key agreement protocol due to Lee, Kim, Kim, & Oh [27]. We present a previously unpublished flaw in the protocol, and demonstrate that the original claims of *known key security resilience* and *key control* are flawed. Finally, a countermeasure is presented. Section 4 presents the conclusion.

2 The BR93 Model

In this section, an informal overview of the BR93 model is provided primarily for the benefit of the reader who is unfamiliar with the model. For a more comprehensive description, the reader is referred to the original paper [6].

The BR93 model defines provable security for entity authentication and key distribution goals. The adversary $\mathcal{A}$ in the model, is a probabilistic machine that controls all the communications that take place between parties by interacting with a set of $\Pi^i_{U_1,U_2}$ oracles ($\Pi^i_{U_1,U_2}$ is defined to be the i^{th} 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). The predefined oracle queries are described informally as follows.

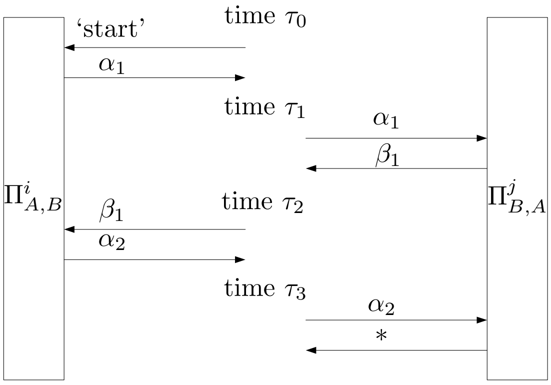
- The $\mathsf{Send}(U_1, U_2, i, m)$ query allows $\mathcal{A}$ to send some message m of her choice to either the client $\Pi^i_{U_1,U_2}$ at will. $\Pi^i_{U_1,U_2}$, upon receiving the query, will compute what the protocol specification demands and return to $\mathcal{A}$ the response message and/or decision. If $\Pi^i_{U_1,U_2}$ has either accepted with some session key or terminated, this will be made known to $\mathcal{A}$.
- The $\mathsf{Reveal}(U_1, U_2, i)$ query allows $\mathcal{A}$ to expose an old session key that has been previously accepted. $\Pi^i_{U_1,U_2}$, upon receiving this query and if it has accepted and holds some session key, will send this session key back to $\mathcal{A}$.
- The $\mathsf{Corrupt}(U_1, K_E)$ 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). This query can be used to model the real world scenarios of an insider cooperating with the adversary or an insider who has been completely compromised by the adversary.
- The $\mathsf{Test}(U_1, U_2, i)$ 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.

Note that in the original BR93 model, the $\mathsf{Corrupt}$ query is not allowed. However, such a query is important as it captures the notion of unknown key share attack [18] and insider attack. Hence, later proofs of security in the BR93 model [2, 8, 9, 13, 17, 28, 29, 35] allow such a query.

2.1 Definition of Partnership

Partnership is defined using the notion of matching conversations, where a conversation is defined to be the sequence of messages sent and received by an oracle. The sequence of messages exchanged (i.e., only the Send oracle queries) are recorded in the transcript, T. At the end of a protocol run, T will contain the record of the Send queries and the responses as shown in Figure 1. Definition 1 gives a simplified definition of matching conversations for the case of the protocol shown in Figure 1.

Definition 1 (BR93 Definition of Matching Conversations [6]). *Let n be the maximum number of sessions between any two parties in the protocol run. Run the protocol shown in Figure 1 in the presence of a malicious adversary $\mathcal{A}$ and consider an initiator oracle $\Pi^i_{A,B}$*



Note that the construction of conversation shown in Definition 1 depends on the number of parties and the number of message flows. Informally, both $\Pi^i_{A,B}$ and $\Pi^j_{B,A}$ are said to be BR93 partners if each one responded to a message that was sent unchanged by its partner with the exception of perhaps the first and last message.

Figure 1: Matching conversation [6]

and a responder oracle $\Pi^j_{B,A}$ who engage in conversations C_A and C_B respectively. $\Pi^i_{A,B}$ and $\Pi^j_{B,A}$ are said to be partners if they both have matching conversations, where

$$\begin{aligned} C_A &= (\tau_0, '\mathit{start}', \alpha_1), (\tau_2, \beta_1, \alpha_2) \\ C_B &= (\tau_1, \alpha_1, \beta_1), (\tau_3, \alpha_2, *), \text{ for } \tau_0 < \tau_1 < \ldots \end{aligned}$$

The matching conversations play a significant role as they bind together incoming and outgoing messages, and uniquely identify a particular session.

2.2 Definition of Freshness

The notion of 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 in the BR93 model, which depends on the notion of partnership in Definition 1.

Definition 2 (Definition of Freshness). *Oracle $\Pi^i_{A,B}$ is fresh (or it holds a fresh session key) at the end of execution, if, and only if, oracle $\Pi^i_{A,B}$ has accepted with or without a partner oracle $\Pi^j_{B,A}$, both oracle $\Pi^i_{A,B}$ and its partner oracle $\Pi^j_{B,A}$ (if such a partner oracle exists) have not been sent a* Reveal *query, and the principals A and B of oracles $\Pi^i_{A,B}$ and $\Pi^j_{B,A}$ (if such a partner exists) have not been sent a* Corrupt *query.*

2.3 Definition of Security

Security 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 simulation $\mathcal{G}$, whose setting is described in Figure 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

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

We require the definition of a negligible function, as described in Definition 3.

Definition 3 ([3]). *A function $\epsilon(k) : \mathbb{N} \to \mathbb{R}$ in the security parameter k, is called negligible if it approaches zero faster than the reciprocal of any polynomial. That is, for every $c \in \mathbb{N}$ there is an integer k_c such that $\epsilon(k) \leq k^{-c}$ for all $k \geq k_c$.*

Definition 4 describes the BR93 security definition.

Definition 4 (BR93 Definition of Security [6]). *A protocol is secure in the BR93 model if for all PPT adversaries $\mathcal{A}$,*

1) *if uncorrupted oracles $\Pi^i_{A,B}$ and $\Pi^j_{B,A}$ complete with matching conversations, then the probability that there exist i, j such that $\Pi^i_{A,B}$ accepted and there is no $\Pi^j_{B,A}$ that had engaged in a matching session is negligible.*

2) $\mathsf{Adv}^{\mathcal{A}}(\mathsf{k})$ *is negligible.*

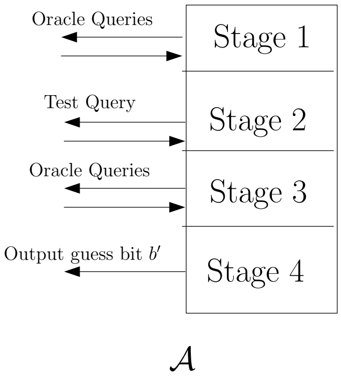
If both requirements of Definition 4 are satisfied, then $\Pi^i_{A,B}$ and $\Pi^j_{B,A}$ will also have the same session key.

2.4 Additional Notions

In order to help the descriptions later we here introduce another property which is often ignored.

Definition 5 (Key Integrity [21]). *Key integrity is the property that the key has not been modified by the adversary, or equivalently only has inputs from legitimate principals.*

- *For a key transport protocol, key integrity means that if the key is accepted by any principal it must be the same key as chosen by the key originator.*



Stage 1: $\mathcal{A}$ is able to send any Send, Reveal, and Corrupt 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. Note that the test session chosen must be fresh. 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.

Stage 3: $\mathcal{A}$ continues interacting with the protocol by making any Send, Reveal, and Corrupt oracle queries of its choice.

Stage 4: Eventually, $\mathcal{A}$ terminates the game simulation and outputs a bit b', which is its guess of the value of b.

Figure 2: Game simulation $\mathcal{G}$

- *For a key agreement protocol, key integrity means that if a key is accepted by any principal it must be a known function of only the inputs of the protocol principals.*

The key replicating attack defined by Krawczyk [26] is described in Definition 6.

Definition 6 (Key Replicating Attack [26]). *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 Case Study

In this section, we present the necessary mathematical preliminaries. We then revisit the two-party ID-based key agreement protocol due to Lee, Kim, Kim, & Oh [27] in this section. We then reveal previously unpublished flaw in the protocol.

3.1 Mathematical Preliminaries

Using the notation of Boneh & Franklin [10], we let $\mathbb{G}_1$ be an additive group of prime order q and $\mathbb{G}_2$ be a multiplicative group of the same order q. We assume the existence of a map $\hat{e}$ from $\mathbb{G}_1 \times \mathbb{G}_1$ to $\mathbb{G}_2$. Typically, $\mathbb{G}_1$ will be a subgroup of the group of points on an elliptic curve over a finite field, $\mathbb{G}_2$ will be a subgroup of the multiplicative group of a related finite field and the map $\hat{e}$ will be derived from either the Weil or Tate pairing on the elliptic curve[1]. The mapping $\hat{e}$ must be efficiently computable and has the following properties.

Bilinearity. For $Q, W, Z \in \mathbb{G}_1$, both $\hat{e}(Q, W + Z) = \hat{e}(Q, W) \cdot \hat{e}(Q, Z)$ and $\hat{e}(Q + W, Z) = \hat{e}(Q, Z) \cdot \hat{e}(W, Z)$.

Non-Degeneracy. For some elements $P, Q \in \mathbb{G}_1$, we have $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$.

Computability. For some elements $P, Q \in \mathbb{G}_1$, we have an efficient algorithm to compute $\hat{e}(P, Q)$.

A bilinear map, $\hat{e}$, is said to be an *admissible* bilinear map if it satisfies all three properties. Since $\hat{e}$ is bilinear, the map $\hat{e}$ is also symmetric.

3.2 Lee–Kim–Kim–Oh ID-Based Key Agreement Protocol

Figure 3 describes the ID-Based key agreement protocol of Lee, Kim, Kim, & Oh [27]. There are two entities in the protocols, namely initiator, A, and responder, B. Both A and B acquired their respective private keys from two cross-domain public key generator environments, PKG_1 and PKG_2. The notation used in the protocol is as follows:

- $\mathcal{H}_1^1$ and $\mathcal{H}_2^1$ denote the secure hash functions used by PKG_1,
- $\mathcal{H}_1^2$ and $\mathcal{H}_2^2$ denote the secure hash functions used by PKG_2,
- $\mathcal{H}$ denotes a common secure hash function used by both PKG_1 and PKG_2,
- $(P_{Pub}^1 = s^1 P^1, s^1)$ and $(P_{Pub}^2 = s^2 P^2, s^2)$ denote the public/private key pairs of PKG_1 and PKG_2 respectively,
- $(Q_A^1, S_A^1 = s^1 Q_A^1)$ denotes the public/private key pair of A acquired from PKG_1,
- $(Q_A^2, S_A^2 = s^2 Q_A^2)$ denotes the public/private key pair of A acquired from PKG_2, and
- $a \in_R \mathbb{Z}_{q1}^*$ and $b \in_R \mathbb{Z}_{q2}^*$ denote the ephemeral private keys of A and B respectively[2]

[1] We note that Tate pairing seems to be more computationally efficient than Weil pairing [20, 22].

[2] Note that a^1 and a^2 shown in the actual paper are the same (i.e., $a^1 = a^2 = a$), which is evident from their discussion in Section 4.2.

$$\begin{array}{ccc} A & & B \\ a \in_R \mathbb{Z}^*_{q1} & & \\ T_{AB} = aP^2 & & \\ W_A = aP^1_{Pub} & \overset{T_{AB}, W_A}{\longrightarrow} & b \in_R \mathbb{Z}^*_{q2} \\ & & T_{BA} = bP^1 \\ & \overset{T_{BA}, W_B}{\longleftarrow} & W_B = bP^2_{Pub} \end{array}$$

Computation of partial session keys

$$\begin{array}{cc} K^1_{AB} = \hat{e}^1(aS^1_A, T_{BA}) & K^1_{BA} = \hat{e}^1(bQ^1_A, W_A) \\ K^2_{AB} = \hat{e}^2(aQ^2_B, W_B) & K^2_{BA} = \hat{e}^2(bS^2_A, T_{AB}) \end{array}$$

Computation of actual session keys

$$\begin{array}{cc} SK_{AB} = \mathcal{H}(\mathcal{H}^1_2(K^1_{AB}), \mathcal{H}^2_2(K^2_{AB})) & SK_{BA} = \mathcal{H}(\mathcal{H}^1_2(K^1_{BA}), \mathcal{H}^2_2(K^2_{BA})) \end{array}$$

Figure 3: Lee–Kim–Kim–Oh ID-based key agreement protocol

For completeness, we will replicate the computation of the partial session keys in order to demonstrate that the actual session key agreed in the absence of a malicious adversary, are indeed the same.

$$\begin{aligned} K^1_{AB} &= \hat{e}^1(aS^1_A, T_{BA}) \\ &= \hat{e}^1(as^1Q^1_A, bP^1) \\ &= \hat{e}^1(Q^1_A, P^1)^{abs^1} \\ &= \hat{e}^1(bQ^1_A, as^1P^1) \\ &= \hat{e}^1(bQ^1_A, aP^1_{Pub}) \\ &= \hat{e}^1(bQ^1_A, W_A) \\ &= K^1_{BA} \end{aligned}$$

$$\begin{aligned} K^2_{AB} &= \hat{e}^2(aQ^2_B, W_B) \\ &= \hat{e}^2(aQ^2_B, bP^2_{Pub}) \\ &= \hat{e}^2(Q^2_B, P^2)^{abs^2} \\ &= \hat{e}^2(bs^2Q^2_B, aP^2) \\ &= \hat{e}^2(bS^2_B, T_{AB}) \\ &= K^2_{BA} \end{aligned}$$

Since $K^1_{AB} = K^1_{BA}$ and $K^2_{AB} = K^2_{BA}$, $SK_{AB} = \mathcal{H}(\mathcal{H}^1_2(K^1_{AB}), \mathcal{H}^2_2(K^2_{AB})) = SK_{BA}$.

3.3 Security Attributes

The protocol described in Figure 3 claims to have the following security attributes.

Known (Session) Key Security. It is often reasonable to assume that the adversary will be able to obtain session keys from any session different from the one under attack. A protocol has known-key security if it is secure under this assumption. This is generally regarded as a standard requirement for key establishment protocols. As Blake-Wilson, Johnson & Menezes [8] have indicated, the Reveal query in the BR93 model is designed to capture the notion of known key security.

Unknown Key-Share Security. Sometimes the adversary may be unable to obtain any useful information about a session key, but can deceive the protocol principals about the identity of the peer entity. Such an attack is first described by Diffie, van Oorschot, & Wiener in 1992 [18], and can result in principals giving away information to the wrong party or accepting data as coming from the wrong party.

As discussed by Boyd & Mathuria [11, Chapter 5.1.2], $\mathcal{A}$ need not obtain the session key to profit 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 this credit deposit as his. Also, a malicious adversary, $\mathcal{A}$, can exploit such an attack in a number of ways if the established session key is subsequently used to provide confidentality (e.g., in AES) or integrity [23]. Consequently security against unknown key-share attacks is regarded as a standard requirement.

Protocols proven secure in the BR93 model that allow the Corrupt query are also proven secure against the unknown-key share attack: that is if a key is to be shared between some parties, U_1 and U_2, the corruption of some other (non-related) player in the protocol, say U_3, should not expose the session key shared between U_1 and U_2 as described by Choo, Boyd, & Hitchcock [16].

Forward Secrecy. When the long-term key of an entity is compromised the adversary will be able to masquerade as that entity in any future protocol runs. However, the situation will be even worse if the adversary can also use the compromised long-term key to obtain session keys that were accepted before the compromise. Protocols that prevent this are said to provide forward secrecy. Since there is usually a computational cost in providing forward secrecy it is sometimes sacrificed in the interest of efficiency.

Forward secrecy for identity-based (ID-based) protocols is similar to conventional public key cryptography. However, there is an additional concern since the master key of the Key Generation Centre (KGC) is another secret that could become compromised. When this happens it is clear that the long-term keys of all users will be compromised. It is possible that a protocol can provide forward secrecy in the usual sense but still give away old session keys if the master key becomes known. We say that a protocol that retains confidentiality of session keys even when the master key is known provides *KGC forward secrecy.*

Key Compromise Impersonation Resistance. Another problem that may occur when the long-term key of an entity A is compromised is that the adversary may be able to masquerade not only *as* A but also *to* A as another party B. Such a protocol is said to allow key compromise impersonation. Resistance to such attacks is often seen as desirable.

(Joint) Key Control. In a key agreement protocol, it is usually desired that no involving entity is able to choose or influence the value of the shared (session) key. This prevents any involving entity from forcing the use of an old key and the non-uniform distribution of the session key.

3.4 A Key Replicating Attack

Figure 4 describes the execution of the two-party ID-based key agreement of Lee, Kim, Kim, & Oh [27] in the presence of a malicious adversary, $\mathcal{A}$.

At the end of the protocol execution shown in Figure 4, both A and B have accepted the same session key, as shown below.

$$\begin{aligned}
K^1_{AB} &= \hat{e}^1(aS^1_A, T_{BA} \cdot c) \\
&= \hat{e}^1(as^1Q^1_A, cbP^1) \\
&= \hat{e}^1(Q^1_A, P^1)^{abcs^1} \\
&= \hat{e}^1(bQ^1_A, acs^1P^1) \\
&= \hat{e}^1(bQ^1_A, acP^1_{Pub}) \\
&= \hat{e}^1(bQ^1_A, W_A \cdot c) \\
&= K^1_{BA} \\
K^2_{AB} &= \hat{e}^2(aQ^2_B, W_B \cdot c) \\
&= \hat{e}^2(aQ^2_B, bcP^2_{Pub}) \\
&= \hat{e}^2(Q^2_B, P^2)^{abcs^2} \\
&= \hat{e}^2(bs^2Q^2_B, acP^2) \\
&= \hat{e}^2(bS^2_B, T_{AB} \cdot c) \\
&= K^2_{BA} \\
SK_{AB} &= \mathcal{H}(\mathcal{H}^1_2(K^1_{AB}), \mathcal{H}^2_2(K^2_{AB})) \\
&= SK_{BA}
\end{aligned}$$

However, both A and B are non-partners since they do not have matching conversations as described in Definition 1. Hence, A succeeds in forcing the establishment of a session, Π_B, (other than the Test session or its matching session) that has the same key as the Test session (i.e., key-replicating attack as described in Definition 6). Consequently, the adversary, $\mathcal{A}$, is able to trivially expose a fresh session key by asking a Reveal query to either A or B, and has a non-negligible advantage in distinguishing the Test key (i.e. $\mathsf{Adv}_{\mathcal{A}}(k)$ is non-negligible).

Key Control Claims. Session keys accepted by both A and B comprise keying material contributed by $\mathcal{A}$, c, in violation of the key integrity property described in Definition 5. In other words, the key control claim by the protocol designers is flawed since the adversary, $\mathcal{A}$, has a non-negligible advantage in coercing the protocol participants into sharing a key when the key is not being shared with $\mathcal{A}$.

Further Remarks. It is trivial to see that the attack revealed in Figure 4 extends to their tripartite protocol, which is an extension of the two-party case. We leave the attack on their extended tripartite protocol as an exercise to interested reader.

3.5 A Counter-Measure

Recent work of Choo, Boyd, & Hitchcock [15, 17] suggests that the inclusion of the sender's and responder's identities and messages sent and received (i.e., $\mathcal{T}_U$ – the concatenation of all messages sent and received) in the key derivation function effectively binds the session key to all messages sent and received by both A and B, as shown below:

$$\begin{aligned}
SK_{AB(Fixed)} &= \mathcal{H}(A||B||\mathcal{T}_A||(\mathcal{H}^1_2(K^1_{AB}), \mathcal{H}^2_2(K^2_{AB}))) \\
SK_{BA(Fixed)} &= \mathcal{H}(A||B||\mathcal{T}_B||(\mathcal{H}^1_2(K^1_{BA}), \mathcal{H}^2_2(K^2_{BA}))) \\
&= SK_{AB(Fixed)},
\end{aligned}$$

If the adversary changes any of the messages in the transmission, the session key will also be different. Intuitively, the attack shown in Figure 4 will no longer be valid, since

$$\begin{aligned}
SK_{AB(Fixed)} &= \mathcal{H}(A||B||\mathcal{T}_A||(\mathcal{H}^1_2(K^1_{AB}), \mathcal{H}^2_2(K^2_{AB}))) \\
SK_{BA(Fixed)} &= \mathcal{H}(A||B||\mathcal{T}_B||(\mathcal{H}^1_2(K^1_{BA}), \mathcal{H}^2_2(K^2_{BA}))) \\
&\neq SK_{AB(Fixed)},
\end{aligned}$$

where

$$\begin{aligned}
\mathcal{T}_A &= T_{AB}||W_A||T_{BA} \cdot c||W_B \cdot c \\
\mathcal{T}_B &= T_{AB} \cdot c||W_A \cdot c||T_{BA}||W_B \\
&\neq \mathcal{T}_A.
\end{aligned}$$

A		$\mathcal{A}$		B
$a \in_R \mathbb{Z}_{q1}^*$		$c \in_R \mathbb{Z}_{q1}^*, \mathbb{Z}_{q2}^*$		
$T_{AB} = aP^2$				
$W_A = aP_{Pub}^1$	$\xrightarrow{T_{AB}, W_A}$		$\xrightarrow{T_{AB} \cdot c, W_A \cdot c}$	$b \in_R \mathbb{Z}_{q2}^*$
				$T_{BA} = bP^1$
	$\xleftarrow{T_{BA} \cdot c, W_B \cdot c}$		$\xleftarrow{T_{BA}, W_B}$	$W_B = bP_{Pub}^2$
		Computation of partial session keys		
$K_{AB}^1 = \hat{e}^1(aS_A^1, T_{BA} \cdot c)$				$K_{BA}^1 = \hat{e}^1(bQ_A^1, W_A \cdot c)$
$K_{AB}^2 = \hat{e}^2(aQ_B^2, W_B \cdot c)$				$K_{BA}^2 = \hat{e}^2(bS_A^2, T_{AB} \cdot c)$
		Computation of actual session keys		
$SK_{AB} = \mathcal{H}(\mathcal{H}_2^1(K_{AB}^1), \mathcal{H}_2^2(K_{AB}^2))$				$SK_{BA} = \mathcal{H}(\mathcal{H}_2^1(K_{BA}^1), \mathcal{H}_2^2(K_{BA}^2))$

Figure 4: Execution of Lee–Kim–Kim–Oh ID-based key agreement protocol in the presence of a malicious adversary, $\mathcal{A}$

4 Conclusion

Through a detailed study of the two-party ID-based authenticated key agreement protocol of Lee, Kim, Kim, & Oh [27], we revealed previously unpublished flaw in the protocol whose purported security is based on heuristic security arguments. Finally, we present a countermeasure due to Choo, Boyd, & Hitchcock [15, 17].

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.

We conclude that designing and analysis correct and secure key agreement protocols remains a hard problem. One (hard) way of obtaining a correct and secure protocol is to provide a complete and detailed proof specification (considering all possible scenarios), rather than providing an informal security analysis or sketchy proofs. We may speculate that the flaws in these three protocols could have been discovered by the protocol designers if complete proof specifications had been constructed.

Acknowledgement

This work was partially funded by the Australian Research Council Discovery Project Grant DP0345775.

References

[1] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Public Key Cryptography - PKC 2005*, LNCS 3386, pp. 65–84, Springer-Verlag, 2005.

[2] S. S. Al-Riyami and K. G. Paterson, "Tripartite authenticated key agreement protocols from pairings," in *9th IMA Conference on Cryptography and Coding*, LNCS 2898, pp. 332–359, Springer-Verlag, 2003.

[3] M. Bellare, "A note on negligible functions," *Journal of Cryptology*, vol. 15, no. 4, pp. 271–284, 2002.

[4] M. Bellare, R. Canetti, and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols," in *30th ACM Symposium on the Theory of Computing - STOC 1998*, pp. 419–428, ACM Press, 1998.

[5] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Eurocrypt 2000*, LNCS 1807, pp. 139 – 155, Springer-Verlag, 2000.

[6] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Crypto 1993*, LNCS 773, pp. 110–125, Springer-Verlag, 1993.

[7] M. Bellare and P. Rogaway, "Provably secure session key distribution: The three party case," in *27th ACM Symposium on the Theory of Computing - STOC 1995*, pp. 57–66, ACM Press, 1995.

[8] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," in *6th IMA International Conference on Cryptography and Coding*, LNCS 1355, pp. 30–45. Springer-Verlag, 1997.

[9] S. Blake-Wilson and A. Menezes, "Security proofs for entity authentication and authenticated key transport protocols employing asymmetric techniques," in *Security Protocols Workshop*, LNCS 1361, pp. 137–158, Springer-Verlag, 1997.

[10] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 585–615, 2003.

[11] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*, Springer-Verlag, June 2003.

[12] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels (extended version available from `http://eprint.iacr.org/2001/040/`)," in *Eurocrypt 2001*, LNCS 2045, pp. 453–474, Springer-Verlag, 2001.

[13] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings (corrected version at `http://eprint.iacr.org/2002/184/`)," in *16th IEEE Computer Security Foundations Workshop - CSFW 2003*, pp. 219–233, IEEE Computer Society Press, 2003.

[14] K. K. R. Choo, "The provably-secure key establishment and mutual authentication protocols lounge, `http://sky.fit.qut.edu.au/~choo/lounge.html`," 2005.

[15] K. K. R. Choo, C. Boyd, and Y. Hitchcock, "Errors in computational complexity proofs for protocols (available from `http://sky.fit.qut.edu.au/~choo/publication.html`)," in *Asiacrypt 2005*, LNCS 3788, pp. 624-643, Springer-Verlag, 2005.

[16] K. K. R. Choo, C. Boyd, and Y. Hitchcock, "Examining indistinguishability-based proof models for key establishment protocols (extended version available from `http://eprint.iacr.org/2005/270`)," in *Asiacrypt 2005*, LNCS 3788, pp. 585-604, Springer-Verlag, 2005.

[17] K. K. R. Choo, C. Boyd, and Y. Hitchcock, "On session key construction in provably secure protocols (extended version available from `http://eprint.iacr.org/2005/206`)," in *1st International Conference on Cryptology in Malaysia - Mycrypt 2005*, LNCS 3715, pp. 116–131, Springer-Verlag, 2005.

[18] W. Diffie, P. C. van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchange," *Journal of Designs, Codes and Cryptography*, vol. 2, pp. 107–125, 1992.

[19] C. J. Fidge, "A survey of verification techniques for security protocols," Technical report 01-22, Software Verification Research Centre, The University of Queensland, Brisbane, 2001.

[20] S. D. Galbraith, "Implementing the tate pairing," in *5th International Symposium on Algorithmic Number Theory - ANTS-V 2002*, LNCS 2369, pp. 324–337, Springer-Verlag, 2002.

[21] P. Janson and G. Tsudik, "Secure and minimal protocols for authenticated key distribution," *Computer Communications*, vol. 18, no. 9, pp. 645–653, 1995.

[22] A. Joux, "The weil and tate pairings as building blocks for public key cryptosystems," in *5th International Symposium on Algorithmic Number Theory - ANTS-V 2002*, LNCS 2369, pp. 20–32, Springer-Verlag, 2002.

[23] B. S. Kaliski, "An unknown key-share attack on the mqv key agreement protocol," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 3, pp. 275–288, 2001.

[24] N. Koblitz and A. Menezes, "Another look at "provable security",". Technical report CORR 2004-20, Centre for Applied Cryptographic Research, University of Waterloo, Canada, 2004.

[25] N. Koblitz and A. Menezes, "Another look at "provable security",". Cryptology ePrint Archive, Report 2004/152, 2004. `http://eprint.iacr.org/2004/152/`.

[26] H. Krawczyk, "HMQV: A high-performance secure diffie-hellman protocol (extended version available from `http://eprint.iacr.org/2005/176/`)," in *Crypto 2005*, LNCS 3621, pp. 546–566, Springer-Verlag, 2005.

[27] H. Lee, D. Kim, S. Kim, and H. Oh, "Identity-based key agreement protocols in a multiple pkg environment," in *International Conference On Computational Science And Its Applications - ICCSA 2005*, LNCS 3483, pp. 877–886, Springer-Verlag, 2005.

[28] P. D. MacKenzie and R. Swaminathan, "Secure network authentication with password identification,". Submitted to the IEEE P1363 Working Group, 1999.

[29] N. McCullagh and Paulo S. L. M. Barreto, "A new two-party identity-based authenticated key agreement (extended version available from `http://eprint.iacr.org/2004/122/`)," in *Cryptographers' Track at RSA Conference - CT-RSA 2005*, LNCS 3376, pp. 262–274, Springer-Verlag, 2005.

[30] C. Meadows, "Open issues in formal methods for cryptographic protocol analysis," in *DARPA Information Survivability Conference and Exposition*, vol. 2052, pp. 237–250, 2000.

[31] C. Meadows, "Formal methods for cryptographic protocol analysis: Emerging issues and trends," *IEEE Journal on Selected Area in Communications*, vol. 21, no. 1, pp. 44–54, 2003.

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

[33] V. Shoup, "On formal models for secure key exchange (ver. 4)," Technical Report RZ 3120 (#93166), IBM Research, Zurich, 1999.

[34] V. Shoup, "OAEP reconsidered," in *Crypto 2001*, LNCS 2139, pp. 239–259, Springer-Verlag, 2001.

[35] D. S. Wong and A. H. Chan, "Efficient and mutually authenticated key exchange for low power computing devices," in *Asiacrypt 2001*, LNCS 2248, pp. 172–289, Springer-Verlag, 2001.

Kim-Kwang Raymond Choo is currently a full-time Ph.D. candidate with Information Security Institute, Queensland University of Technology, Australia; and a part-time MBA student with the University of Queensland, Australia. He received the BSc Maths, BAppSci (Hons) Industrial & Applied Maths, and Master of Information Technology degrees in Dec 2000, Dec 2002, and May 2002 respectively. His research interests include formal specification and analysis of mutual authentication and/or key establishment protocols, and provably-secure protocols. He has served as a program committee member and an external reviewer for several international conferences and journals.