Relations among Security Models for Authenticated Key Exchange

Jeong Ok Kwon and Ik Rae Jeong

Usually, key-establishment protocols are suggested in a security model. However, there exist several different security models in the literature defined by their respective security notions. In this paper, we study the relations between the security models of key establishment. For the chosen security models, we first show that some proven key-establishment protocols are not secure in the more restricted security models. We then suggest two compilers by which we can convert a key-establishment protocol that is secure in a specific security model into a key-establishment protocol that is still secure in a more restricted security model.

Keywords: Key establishment, security model, compiler, HMQV, KAM.

I. Introduction

When we design a cryptographic protocol, we should prove the security of the protocol in a security model. A security model is designed to reflect attacks in the real world.

In the field of key establishment, there are many security requirements, and there are many key-establishment protocols that are trying to achieve some of these security requirements. However, there are only a few key-establishment protocols achieving most of these security requirements (strong key-establishment protocols). Unfortunately, these strong key-establishment protocols were proven only in newer security models. Therefore, we do not know whether such strong key-establishment protocols are also secure in more traditional security models.

Bellare and Rogaway suggested the first formal security model for key exchange — a security model called the BR model [1]. The BR model assumes the network communication channel to be a half-duplex channel. In a half-duplex channel, the two communicating parties have to send their messages alternatively. A less restricted security model, one that perhaps makes it easier to construct a strong key-establishment protocol, for a half-duplex channel is suggested by Krawczyk, which we call the KR model [2]. In a duplex channel, the two communicating parties can send their messages at the same time. For such a channel, it is necessary to create new security models since those for the half-duplex channel are not transferable; one such model is defined by Jeong and others, which we call the JKL model [3].

1. BR Model

In this paper, we consider only key-establishment protocols

Manuscript received Oct. 25, 2013; revised Mar. 5, 2014; accepted May 29, 2014.

This work was supported by Basic Science Research Programs through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2013R1A2A2A01068200).

Jeong Ok Kwon (jeongokkwon@gmail.com) is with the Samsung SDS, Seoul, Rep. of Korea.

Ik Rae Jeong (corresponding author, irjeong@korea.ac.kr) is with the Graduate School of Information Security, Korea University, Seoul, Rep. of Korea.

that are for use with two parties — namely, two-party key-establishment protocols. In [1], Bellare and Rogaway suggested the BR model for two-party key-establishment protocols that are used in a half-duplex channel. In a two-party key-establishment protocol, two parties establish a common session key between them. One of the most basic two-party key-establishment protocols is the Diffie–Hellman key-establishment protocol [4]. In the basic unauthenticated Diffie–Hellman key-establishment protocol, an initiator of the protocol first sends g^{α_i} to a responder, where g is a generator of a group and α_i is the random value selected by the initiator. After receiving g^{α_i}, the responder sends g^{α_j} to the initiator. Then the initiator and the responder calculate a session key, $sk = g^{\alpha_i\alpha_j}$. In the basic Diffie–Hellman key-establishment protocol, an initiator and a responder alternatively send their messages through their network communication channel. In a half-duplex channel, the communicating parties cannot send their messages at the same time. Most two-party key-establishment protocols are constructed for use in a half-duplex channel. The basic unauthenticated Diffie–Hellman key-establishment protocol for a half-duplex channel is depicted as follows:

	P_i (initiator)	P_j (responder)
Round 1	g^{α_i}	
Round 2		g^{α_j}

The session key is $sk = g^{\alpha_i\alpha_j}$.

2. KR Model

In [2], Krawczyk suggested the KR model for a half-duplex channel. In this model, an initiator and a responder exchange their messages alternatively, as in the BR model. However, in the KR model, it does not matter which of the two communicating parties is the initiator. That is, we can ignore the roles of initiator and responder in the KR model. To see the difference between the two models, consider an adversary A who behaves as follows on the above unauthenticated Diffie–Hellman protocol: (a) A makes P_i initiate a run of a protocol between P_i and P_j and receives g^{α_i} from P_i, (b) A makes P_j initiate another run of a protocol between P_i and P_j and receives g^{α_j} from P_j, (c) A sends g^{α_i} to P_j in the latter run and g^{α_j} to P_i in the former run. As a result of the behavior of A, P_i and P_j obtain the same protocol messages g^{α_i} and g^{α_j} in the two runs, and thus generate the same session key $sk = g^{\alpha_i\alpha_j}$. In addition, P_i thinks that they are the initiator and P_j thinks that they are the initiator. This situation is depicted as follows:

P_i's view:

	P_i (initiator)	P_j (responder)
Round 1	g^{α_i}	
Round 2		g^{α_j}

P_j's view:

	P_i (responder)	P_j (initiator)
Round 1		g^{α_j}
Round 2	g^{α_i}	

The BR model considers the above attack as dangerous; hence, for a key-establishment protocol to be secure, it must be able to prevent such an attack. In contrast, the KR model does not consider the above attack to be dangerous; consequently, the key-establishment protocol is not required to prevent such an attack. In other words, the above run of a protocol is considered as an "intact" run in the KR model, while it is not considered intact in the BR model. The intact run is defined using a notion, partnered (or matched), in the formal security model. For the above run of a protocol, P_i and P_j have the notion of being partnered in the KR model, whereas P_i and P_j do not in the BR model.

3. JKL Model

In [3], Jeong and others suggested the JKL model for a duplex channel. In a duplex channel, the two communicating parties can send their messages simultaneously. The basic unauthenticated Diffie–Hellman key-establishment protocol in a duplex channel is depicted as follows:

	P_i	P_j
Round 1	g^{α_i}	g^{α_j}

The session key is $sk = g^{\alpha_i\alpha_j}$.

In the above key-establishment protocol, one of the two communicating parties sends g^{α_i}; while, at the same time, the other sends g^{α_j} without waiting for the communicating partner's message. Note that the basic Diffie–Hellman key-establishment protocol requires two rounds in a half-duplex channel but only one round in a duplex channel. Thus, conducting a key-establishment protocol in a duplex channel is more efficient.

4. Authenticated Key Establishment

The basic Diffie–Hellman key-establishment protocol does not provide authentication of the session keys. That is, an adversary can impersonate the communicating party and send messages of its choice to the target party. As a result, the

adversary can generate the same session key as the target party. There are many schemes (called authenticated key-establishment protocols) that provide authentication of session keys. Authentication of a session key simply means that no party, besides those involved in the communication, can obtain the session key information. Authenticated key-establishment protocols serve as basic building blocks for constructing secure, complex, and higher-level cryptographic protocols. In this paper, we consider only authenticated key-establishment protocols where each party holds a pair of private/public keys.

A party typically uses its private key and randomly generated numbers to establish a session key. It is easy to observe that without the use of the randomly generated numbers, all session keys would be identical. For secure subsequent use of a session key, most applications require that each session key be unique. Hence, to establish a new session key, it is necessary to use uniquely different randomly generated numbers along with the same private key each time. A private key is also used to authenticate a session key.

Several security notions pertinent to key establishment have been defined based on considerations of how to minimize the damage caused by an adversary acquiring either of the private keys, the random numbers used, or the session keys. One of the security notions considered most often in the literature is that of known-key security (KKS). Protocols providing KKS are secure against Denning–Sacco attacks [5] that aim at trying to compromise multiple session keys (for sessions other than those that require a guarantee of secrecy).

Forward secrecy considers scenarios in which an adversary is able to obtain the private keys of some parties. There are two kinds of forward secrecy: weak and strong. Protocols achieving weak forward secrecy (w-FS) maintain the secrecy of a session key that is shared through an execution of the protocol without any interference by an adversary. Protocols achieving strong forward secrecy (s-FS) maintain the secrecy of a session key, even if the session key has been established with the interference of an adversary.

Besides the above security notions, there are others, such as key compromise impersonation (KCI) and unknown key-share (UKS) [6]–[8]. Security against session-state reveal (SSR) is formally considered in [2] and [9].

5. Our works

Most authenticated key-establishment protocols are constructed in the BR model [1], [6], [10]–[13]. The standardization of key establishment has been done in the BR model [14]–[18]. We note that a comparison of the original BR model and subsequent variants of it have been analyzed in [19].

Some key-establishment schemes are proved in the KR model [2], [20]. There exist a few key-establishment schemes in the JKL model [3], [21]–[22]. We note that a multiparty authenticated key-establishment protocol can be used as a two-party authenticated key-establishment protocol. Thus, a multiparty key-establishment protocol in a broadcasting channel model, such as that in [21] and [23]–[24], can be used as a two-party key-establishment protocol in the JKL model.

The strong key-establishment protocols FS, KCI, and SSR, achieving most of the known security notions, are known respectively as HMQV [2], Okamoto's scheme [20], and KAM [22]. HMQV and Okamoto's scheme are secure in the KR model, and KAM is secure in the JKL model. HMQV is proven in the random oracle model, whereas Okamoto's scheme and KAM are proven in the standard model. HMQV is more efficient than both KAM and Okamoto's scheme.

We will show that a key-establishment protocol, secure in a security model, might not be secure in a more restricted security model. That is, we cannot simply use a key-establishment protocol that is secure in the KR model or in the JKL model as a key-establishment protocol in the BR model. For example, we will show that both HMQV and KAM are not secure in the BR model.

We will then suggest a compiler that converts a secure key-establishment protocol in the JKL model into a secure key-establishment protocol in the KR model and another compiler that converts a secure key-establishment protocol in the KR model into a secure key-establishment protocol in the BR model. Our compilers do not modify any protocol messages of the underlying key-establishment scheme. Moreover, our compilers do not increase any computational complexity except adding only one pseudorandom function evaluation. Using compilers, we can convert both HMQV and KAM into HMQV_S and KAM_S, respectively, both of which are secure in the BR model.

In Section II, we review the primitives used in the paper. In Section III, we define the BR model, the KR model, and the JKL model. In Section IV, we show that HMQV and KAM are not secure in the BR model. In Section V, we suggest compilers by which we can convert a key-establishment protocol secure in a specific security model into a key-establishment protocol secure in a more restricted model. In Section VI, we conclude the paper.

II. Preliminaries

We use the following notations: (a) $[a, b]$ denotes a set of integers from a to b, (b) $c \leftarrow S$ denotes that c is randomly selected from a set S, (c) a concatenation of two strings a and b is denoted by $a\|b$ and (d) if *evt* is an event, then Pr[*evt*] is used to denote the probability of it occurring.

Let θ be a security parameter. We use a secure pseudorandom function F whose domain and range are $\{0, 1\}^\theta$ [24]. Let F_K: $\{0, 1\}^\theta \rightarrow \{0, 1\}^\theta$ be a function selected from a function family F, where F = $\{F_K|K$ is in the space of θ-bit strings$\}$.

III. Security Model for Key Establishment

We will explain the BR model, KR model, and JKL model in a unified way. We assume that each party's identity is denoted as P_i and that each party holds a pair of private/public keys.

We consider key-establishment protocols in which two parties want to exchange a session key using their public keys to provide key authentication [25]. The kth instance of party P_i is denoted by $\Pi_{i,k}$. If a key-establishment protocol terminates, then $\Pi_{i,k}$ generates a session key $sk_{i,k}$.

A session identifier of an instance $\Pi_{i,k}$, denoted $sid_{i,k}$, is a string that has a high probability of being different from those of all other sessions in the system. We first define $\text{strID}_{i,k}$ and $\text{strTIME}_{i,k}$. T[0][0]he concatenation of all messages sent and received by a particular instance $\Pi_{i,k}$, where the order of these messages is determined by the lexicographic ordering of the two parties' identities, is given by $\text{strID}_{i,k}$. The concatenation of all messages sent and received by a particular instance $\Pi_{i,k}$, where the order of these messages is determined by the time they were sent or received, is given by $\text{strTIME}_{i,k}$.

We note that an adversary can delay or reorder messages. The session identifiers are used to define partnered (or matched) instances in the security models and are appropriately defined as follows:

- A session identifier in the BR model has been defined as $sid_{i,k} = \text{strTIME}_{i,k}$ [1], [26].
- A session identifier in the KR model has been defined as $sid_{i,k} = \text{strID}_{i,k}$ [2].
- A session identifier in the JKL model has been defined as $sid_{i,k} = \text{strID}_{i,k}$ [3], [22]. (Note that $\text{strTIME}_{i,k}$ cannot be used as a session identifier in the JKL model, because the two parties may send their messages simultaneously.)

We note that the security models are different according to how the notion of being "partnered" is defined. As a consequence, the security models are different according to the way the session identifier is defined.

Consider instance $\Pi_{i,k}$ of party P_i. The partner of this instance is the party $P_j (\neq P_i)$ with whom P_i believes it is interacting. We say that the two instances $\Pi_{i,k}$ and $\Pi_{j,k'}$ are partnered (or matching), if the following conditions are true: $sid_{i,k} = sid_{j,k'}$, P_j is the partner of $\Pi_{i,k}$, and P_i is the partner of $\Pi_{j,k'}$.

Note that if two instances are partnered in the BR model, the two instances are also partnered in the KR model.

For a concrete definition of the KKS/w-FS/s-FS/KCI/SSR-security through an experiment, refer to [1]–[3].

IV. Security Analysis of HMQV

It is obvious that a key-establishment protocol secure in the BR model is also secure in the KR model, and a key-establishment protocol secure in the KR model is also secure in the JKL model. In this section, we show that we cannot simply use a key-establishment protocol that is secure in the KR model or in the JKL model as a key-establishment protocol in the BR model.

Let $(x_k, y_k = g^x k)$ be a pair of private/public keys of P_i. Let H be a collision-resistant hash function and Mac a strongly unforgeable MAC-generation function. Then HMQV is defined as follows [2]:

	P_i (initiator)	P_j (responder)
Round 1	g^{α_i}	
Round 2		g^{α_j}

The subsequent session key is calculated as $sk = H(g^{(\alpha_i + dx_i)(\alpha_j + ex_j)})$, where $d = H(g^{\alpha_i} \| y_j)$ and $e = H(g^{\alpha_j} \| y_i)$.

KAM is defined as follows [22]:

	P_i's message	P_j's message
Round 1	g^{α_i}	g^{α_j}
Round 2	τ_i	τ_j

The MAC keys are calculated as $k_{ij} = H(g^{\alpha_j x_i})$ and $k_{j,i} = H(g^{\alpha_i x_j})$. Note that $k_{ij} \neq k_{j,i}$. τ_i and τ_j are MAC values, where $\tau_i \leftarrow \text{Mac}k_{ij}(i\|j\|g^{\alpha_i})$ and $\tau_j \leftarrow \text{Mac}k_{j,i}(j\|i\|g^{\alpha_j})$. The session key is calculated as $sk = H(g^{x_i x_j}) \oplus H(g^{\alpha_i \alpha_j})$.

Theorem 1. HMQV is w-FS/KCI/SSR-secure in the KR model [2].

Theorem 2. HMQV is insecure against known-key attacks in the BR model. That is, HMQV does not provide KKS in the BR model.

Proof of Theorem 2. Consider an adversary A that creates two sessions among two parties P_i and P_j. In the first session, P_i initiates a protocol such that P_i sends g^{α_i}. In the second session, P_j initiates a protocol such that P_j sends g^{α_j}. Then A interleaves protocol messages such that A sends g^{α_j} to P_i in the first session and g^{α_i} to P_j in the second session, as follows:

The first session ($\Pi_{i,1}$'s view)

	P_i (initiator)	P_j (responder)
Round 1	g^{α_i}	
Round 2		g^{α_j}

The second session ($\Pi_{j,1}$'s view)

	P_i (responder)	P_j (initiator)
Round 1		g^{α_j}
Round 2	g^{α_i}	

Adversary A obtains $sk_{j,1}$, a session key of P_j in the second session, through the Reveal(j, 1) query. Then A knows $sk_{i,1}$, a session key of P_i in the first session, and thus breaks KKS. We can easily determine when $sk_{i,1}$ and $sk_{j,1}$ are the same. Note that $\text{strTIME}_{i,1} = g^{\alpha_i} || g^{\alpha_j}$ and $\text{strTIME}_{j,1} = g^{\alpha_j} || g^{\alpha_i}$. If we assume that ($P_i < P_j$),[1] then $\text{strID}_{i,1} = \text{strID}_{j,1} = g^{\alpha_i} || g^{\alpha_j}$. This attack strategy does not violate the KKS in the BR model because $\Pi_{i,1}$ and $\Pi_{j,1}$ are not partnered. Note that $sid_{i,1}$ (= $\text{strTIME}_{i,1}$) and $sid_{j,1}$ (= $\text{strTIME}_{j,1}$) are different in the BR model. On the other hand, A cannot make a Reveal (j,1) query in the KR model because $\Pi_{i,1}$ and $\Pi_{j,1}$ are partnered. Note that $sid_{i,1}$ (= $\text{strID}_{i,1}$) and $sid_{j,1}$ (= $\text{strID}_{j,1}$) are the same in the KR model. ■

V. Compilers among Security Models

In this section, we suggest two compilers: compiler1 and compiler2. Compiler1 converts a key-establishment protocol in the JKL model into a key-establishment protocol in the KR model. Compiler2 converts a key-establishment protocol in the KR model into a key-establishment protocol in the BR model.

Let KE_D be a secure key-establishment protocol in the JKL model. Then compiler1 converts KE_D into a new key-establishment protocol, KE_W, which is secure in the KR model. This procedure is explained below.

Setup. Let KE_D be a secure key-establishment protocol in the JKL model. Person P_i is going to establish a session key with P_j ($\neq P_i$). Assume that P_i is an initiator and that P_j is a responder. Person P_j behaves as similarly as P_i, so we describe the protocol on behalf of P_i.

Round messages. In the lth round of KE_D, P_i and P_j may send their lth-round messages simultaneously in the JKL model. But in the KR model, only one of the two can send its message in any given round. So, we break the symmetry of KE_D in such a way that initiator P_i first sends its message of the lth round of KE_D in a given round, and then responder P_j sends its message of the lth round of KE_D in the next round. So, if KE_D is an n-round protocol, then KE_W is at most a $2n$-round protocol.

Computation of session keys. Person P_i calculates a session key for KE_W in the same way as a session key for KE_D is calculated in the JKL model.

Theorem 3. If KE_D is XX-secure in the JKL model, then KE_W converted by compiler1, is XX-secure in the KR model, where XX is either KKS, w-FS, s-FS, KCI, or SSR.

Proof of Theorem 3. Let A_W be an adversary against KE_W in the KR model. Then we can construct an adversary A_D against KE_D using A_W in the JKL model, such that the advantages of A_D and A_W are the same. Adversary A_D receives queries from A_W, gets answers from its own oracles, and returns the answers to A_W. Note that A_D perfectly simulates KE_W to A_W, since the two instances are partnered in the JKL model if and only if the two instances are partnered in the KR model. ■

In [22], Jeong and others suggested a key-establishment protocol, KAM, in the duplex channel and proved its security in the JKL model.

Theorem 4. KAM is s-FS/KCI/SSR-secure without random oracles in the JKL model [22].

Let KAM_W be the converted protocol from KAM using compiler1 for the KR model.

Corollary 1. KAM_W is s-FS/KCI/SSR-secure without random oracles in the KR model.

Proof of Corollary 1. From Theorem 3 and Theorem 4, Corollary 1 follows. ■

Even though KAM_W is secure in the KR model, KAM_W is insecure against know-key attacks in the BR model.

Theorem 5. KAM_W is insecure against known-key attacks in the BR model. That is, KAM_W does not provide KKS in the BR model.

Proof of Theorem 5. We can easily see that the same attack in the proof of Theorem 2 also applies for KAM_W. ■

To make a secure key-establishment protocol in the BR model from a secure key-establishment protocol in the KR model, we construct another conversion method; namely, then compiler2. If KE_W is a secure key-establishment protocol in the KR model, compiler2 converts KE_W into KE_S. This procedure is explained below.

Setup. Let KE_W be a secure key-establishment protocol in the KR model, and let F be a secure pseudorandom function. Let $\hbar$ be a collision-resistant hash function such that $\hbar : \{0, 1\}^* \rightarrow \{0, 1\}^\theta$. We assume that the session key space of KE_W and the pseudorandom function key space of F are a space of θ-bit strings. Person P_i is going to establish a session key with P_j ($\neq P_i$). Assume that P_i is an initiator and P_j is a responder. Person P_j behaves as similarly as P_i, so we describe the protocol on behalf of P_i.

Round messages. KE_S's round messages are the same as those of KE_W.

1) We denote P_i precedes P_j in the lexicographic ordering as $P_i < P_j$.

Computation of session keys. P_i calculates sk_{KE_W} in the same way as a session key of KE_W is calculated in the KR model. Person P_i makes the session identifier *sid* = strTIME and session key $sk_{KE_S} = Fsk_{KE_W}(\hbar(sid))$ for KE_S.

Theorem 6. If F is a secure pseudorandom function and KE_W is XX-secure in the KR model, then KE_S converted by compiler2 is XX-secure in the BR model, where XX is either KKS, w-FS, s-FS, KCI, or SSR. More concretely, $Adv^{XX}_{KE_S}(\theta, t) \leqslant (2+2(Nqs)^2) \times Adv^{XX}_{KE_W}(\theta, t) + 4Adv^{RRF}_{F}(\theta, t, 2)$, where t is the maximum total experiment time including an adversary's execution time. Here, N is an upper bound on the number of honest[2] parties, and q_s is an upper bound on the number of the sessions an adversary creates.

Proof of Theorem 6. The intuition is as follows. For an adversary A to get a non-negligible advantage for KE_S, the adversary has to break the underlying key-establishment scheme KE_W or a pseudorandom function F. We show that using A, we can make either an adversary D that breaks KE_W or an E that breaks F. Some difficulties arise when we construct a D that simulates KE_S to A. This is because in some cases A is allowed to make a Reveal query against KE_S but D is not allowed to make a Reveal query against KE_W. We solve this difficulty using the fact that if there exists oracle $\Pi_{j,k'}$ partnered with tested oracle $\Pi_{i,k}$, the advantage of an adversary is the same as in the case when it asks a Test query to $\Pi_{j,k'}$. The details of which are as follows.

Assume that A makes a Test(i, k) query and that $\Pi_{i,k}$'s partner is P_j. The advantage of A then occurs in the following two cases:

- Case 1. There exists an oracle $\Pi_{j,k'}$ such that $strTIME_{j,k'} = strTIME_{i,k}$[3] $\vee$ there exists no oracle $\Pi_{j,k'}$ such that $strID_{j,k'} = strID_{i,k}$.
- Case 2. There exists an oracle $\Pi_{j,k'}$ such that $strID_{j,k'} = strID_{j,k}$ and $strTIME_{j,k'} \neq strTIME_{j,k}$.

We restrict the advantages from the above two cases in the following claims:

- Claim 1. $Adv^{Case(1)}_{KE_S} \leq 2Adv_{KE_W} + 2Adv^{PRF}_{F}$.
- Claim 2. $Adv^{Case(2)}_{KE_S} \leq 2(Nq_s)^2 \times Adv_{KE_W} + 2Adv^{PRF}_{F}$.

From Claim 1 and Claim 2, Theorem 6 follows. Next, we prove the claims.

Proof of Claim 1. Consider an adversary A attacking KE_S with the advantage from Case 1 in the sense of XX-security. We define the following three games:

- In $Game_0$, a session key for the test query is calculated and returned to the adversary as follows: if $b = 1$, $sk = Fsk_{KE_W}(\hbar(sid))$. If $b = 0$, $sk \leftarrow \{0,1\}^{\theta}$.
- In $Game_1$, a session key for the test query is calculated and returned to the adversary as follows: if $b = 1$, $sk = F_R(\hbar(sid))$, where $R \leftarrow \{0, 1\}^{\theta}$. If $b = 0$, $sk \leftarrow \{0, 1\}^{\theta}$.
- In $Game_2$, a session key for the test query is calculated and returned to the adversary as follows: If $b = 1$, $sk = h(\hbar(sid))$, where $h \leftarrow Rand^{\{0,1\}^{\theta} \rightarrow \{0,1\}^{\theta}}$. If $b = 0$, $sk \leftarrow \{0,1\}^{\theta}$.

The difference of the advantage of A from Case 1 in the aforementioned games is restricted as follows:

- Claim 1.1. $Adv^{Game_0}_{KE_S,A} - Adv^{Game_1}_{KE_S,A} \leq 2Adv_{KE_W}$.
- Claim 1.2. $Adv^{Game_1}_{KE_S,A} - Adv^{Game_2}_{KE_S,A} \leq 2Adv^{PRF}_{F}$.

It is obvious that the advantage of any adversary is 0 in $Game_2$. Thus, from Claim 1.1 and Claim 1.2, Claim 1 follows. ■

Now, we proceed to prove Claim 1.1 and Claim 1.2.

Proof of Claim 1.1. Consider the following algorithm D that tries to break KE_W using A that tries to break KE_S. Algorithm D receives all oracle queries from A and answers them using its own oracle queries to KE_W. A more concrete description of D is as follows:

1) For each oracle query of A, Algorithm D answers it using its own oracle query to KE_W as follows:
   - For Send(i, k) query: D just asks Send(i, k) query in its own experiment.
   - For Corrupt(i) query: D just asks Corrupt(i) query in its own experiment, receives the private key of P_i, and returns the private key to A.
   - For State(i, k) query: D just asks State(i, k) query in its own experiment, receives the random values of $\Pi_{i,k}$, and returns the random values to A.
   - For Reveal(i, k) query: D gets sk_{KE_W} through its own Reveal query and returns $sk_{KE_S} = Fsk_{KE_W}(\hbar(sid_{i,k}))$ to A.
   - For Test(i, k) query: D gets τ through its own Test query and flips a coin b. If b is equal to 1, D returns $sk = F_{\tau}(\hbar(sid_{i,k}))$ to A. Otherwise, D returns a random value selected from $\{0,1\}^{\theta}$.
2) Assume that A outputs b' then quits. If $b = b'$, D outputs 1. Otherwise, D outputs 0.

Algorithm D simulates $Game_0$ or $Game_1$ depending on whether τ is a real session key of KE_W or not. So, the following inequality holds:

$$\begin{aligned} Adv_{D,KE_W} &= Pr[D() = 1 | \tau = sk_{KE_W}] - Pr[D() = 1 | \tau \leftarrow \{0,1\}^{\theta}] \\ &= Pr_A[b = b' | sk = Fsk_{KE_W}(\hbar(sid))] \\ &\quad - Pr_A[b = b' | sk = F_R(\hbar(sid)); R \leftarrow \{0,1\}^{\theta}] \\ &= 1/2(Adv^{Game_0}_{A} - Adv^{Game_1}_{A}). \end{aligned}$$ ■

Proof of Claim 1.2. If the difference of advanatages of adversary A between $Game_1$ and $Game_2$ is non-negligible, we can construct an algorithm E that breaks the pseudorandomness of F with a non-negligible probability using A. Consider a distinguisher E to break the pseudorandomness of a pseudorandom function family F. Distinguisher E is given an oracle function $f(\cdot)$ in the experiment of pseudorandomness

2) We name the parties that are not insider attacker honest parties.

3) $\Pi_{i,k}$ and $\Pi_{j,k'}$ are partnered.

of the function family F. Distinguisher E uses $f(\cdot)$ to generate a session key for the test oracle. A more concrete description of E is as follows:

1) Distinguisher E is given an oracle function $f(\cdot)$. E simulates Game_1.
2) For each oracle query of A, E answers it as in Game_1, except in the case of Test(i, k) query whereby E flips a coin b. If b is equal to 1, E returns $f(\hbar(sid_{i,k}))$. Otherwise, E returns a random value selected from $\{0,1\}^{\theta}$.
3) Assume that A outputs b' then quits. If $b = b'$, E outputs 1. Otherwise, E outputs 0.

Distinguisher E simulates Game_1 or Game_2 depending on whether $f(\cdot)$ is a function from F. So, the following inequality holds:

$$\begin{aligned}\text{Adv}^{\text{PRF}}{}_E &= \Pr[E^{f(\cdot)} = 1 | K \leftarrow \{0,1\}^{\theta}; f = \text{F}_K] \\ &\quad - \Pr[E^{f(\cdot)} = 1 | h \leftarrow \text{Rand}^{\{0,1\}^{\theta} \rightarrow \{0,1\}^{\theta}}; f = h] \\ &= \Pr_A[b = b' | K \leftarrow \{0,1\}^{\theta}; f = \text{F}_K] \\ &\quad - \Pr_A[b = b' | h \leftarrow \text{Rand}^{\{0,1\}^{\theta} \rightarrow \{0,1\}^{\theta}}; f = h] \\ &= 1/2\,(\text{Adv}^{\text{Game}_1}{}_A - \text{Adv}^{\text{Game}_2}{}_A).\end{aligned}$$ ■

Proof of Claim 2. Consider an adversary A attacking KE_S with the advantage from Case 2 in the sense of XX-security. Let $\Pi_{i^*,t1}$ be the tested oracle. Then there exists an oracle $\Pi_{j^*,t2}$ such that $\text{strID}_{j^*,t2} = \text{strID}_{i^*,t1}$ and $\text{strTIME}_{j^*,t2} \neq \text{strTIME}_{i^*,t1}$. We define the following three games:

- Game_0
  - A session key for Reveal query is calculated and returned to the adversary according to the protocol.
  - A session key for Test query is calculated and returned to the adversary as follows: If $b = 1$, $sk = \text{F}sk_{\text{KE}_W}(\hbar(sid_{i^*,t1}))$. If $b = 0$, $sk \leftarrow \{0,1\}^{\theta}$.
- Game_1
  - A session key of $\Pi_{j^*,t2}$ for Reveal query is asked, $sk = \text{F}_R(\hbar(sid_{j^*,t2}))$ is returned, where $R \leftarrow \{0,1\}^{\theta}$. A session key of $\Pi_{i,k}$ ($\neq \Pi_{j^*,t2}$) for Reveal query is calculated and returned to the adversary as in Game_0.
  - A session key for Test query is calculated and returned to the adversary as follows: If $b = 1$, $sk = \text{F}_R(\hbar(sid_{i^*,t1}))$, where $R \leftarrow \{0,1\}^{\theta}$. If $b = 0$, $sk \leftarrow \{0,1\}^{\theta}$.
- Game_2
  - A session key for Reveal query is calculated and returned to the adversary as in Game_1.
  - A session key for Test query is calculated and returned to the adversary as follows: If $b = 1$, $sk = h(\hbar(sid_{i^*,t1}))$, where $h \leftarrow \text{Rand}^{\{0,1\}^{\theta} \rightarrow \{0,1\}^{\theta}}$. If $b = 0$, $sk \leftarrow \{0,1\}^{\theta}$.

The difference of the advantage of A from Case 2 in the aforementioned games is restricted as follows:

- Claim 2.1. $\text{Adv}^{\text{Game}_0}{}_{\text{KE}_S,A} - \text{Adv}^{\text{Game}_1}{}_{\text{KE}_S,A} \leq 2(Nq_s)^2 \cdot \text{Adv}_{\text{KE}_W}$.
- Claim 2.2. $\text{Adv}^{\text{Game1}}{}_{\text{KE}_S,A} - \text{Adv}^{\text{Game}_2}{}_{\text{KE}_S,A} \leq 2\text{Adv}^{\text{PRF}}{}_{\text{F}}$.

It is obvious that the advantage of any adversary is 0 in Game_2. Thus, from Claim 2.1 and Claim 2.2, Claim 2 follows. ■

Now, we proceed to prove Claim 2.1 and Claim 2.2.

Proof of Claim 2.1. Consider the following algorithm D that tries to break KE_W using A that tries to break KE_S. Algorithm D receives all oracle queries from A and answers them using its own oracle queries to KE_W. Algorithm D first guesses the tested oracle $\Pi_{i,k}$ and an oracle $\Pi_{j,k'}$ such that $\text{strID}_{j,k'} = \text{strID}_{i,k}$ and $\text{strTIME}_{j,k'} \neq \text{strTIME}_{i,k}$. A more concrete description of D is as follows:

1) Algorithm D randomly selects i^* and j^* from $[1,N]$, and t_1, t_2 from $[1, q_s]$. Algorithm D sets two flags — namely, *reveal-then-test* = 0 and *test-then-reveal* = 0.
2) For each oracle query of A, D answers it using its own oracle query to KE_W as follows:
   - For Send(i, k) query: D just asks Send(i,k) query in its own experiment.
   - For Corrupt(i) query: D just asks Corrupt(i) query in its own experiment, receives the private key of P_i, and returns the private key to A.
   - For State(i, k) query: D just asks State(i,k) query in its own experiment, receives the random values of $\Pi_{i,k}$, and returns the random values to A.
   - For Reveal(i,k) query.
     - If $j = j^* \wedge k = t_2$ and *test-then-reveal* = 0: D gets τ_r through its own Test(j^*, t_2) query, and sets *reveal-then-test* = 1. D returns $sk = \text{F}\tau_r(\hbar(sid_{j^*,t2}))$ to A.
     - Else if $i = j^* \wedge k = t_2$ and *test-then-reveal* = 1: D returns $sk = \text{F}\tau_t(\hbar(sid_{j^*,t2}))$ to A, where τ_t was already obtained in Test query.
     - Else D gets sk_{KE_W} through its own Reveal(i,k) query and returns $sk = \text{F}sk_{\text{KE}_W}(\hbar(sid_{i,k}))$ to A.
   - For Test(i,k) query.
     - If $i \neq i^* \vee k \neq t_1$, D fails and stops.
     - If *reveal-then-test* = 0: D gets τ_t through its own Test(i^*, t_1) query and sets *test-then-reveal* = 1. D flips a coin b. If b is equal to 1, D returns $sk = \text{F}\tau_t(\hbar(sid_{i^*,t1}))$ to A. Otherwise, D returns a random value selected from $\{0,1\}^{\theta}$.
     - Else if *reveal-then-test* = 1: D flips a coin b. If b is equal to 1, D returns $sk = \text{F}\tau_r(\hbar(sid_{i^*,t1}))$ to A, where τ_r was already obtained in Reveal query. Otherwise, D returns a random value selected from $\{0,1\}^{\theta}$.
3) Assume that A outputs b' then and quits. If there exists no oracle $\Pi_{j^*,t2}$ such that $\text{strID}_{j^*,t2} = \text{strID}_{i^*,t1}$ and $\text{strTIME}_{j^*,t2} \neq \text{strTIME}_{i^*,t1}$, D fails and stops. If $b = b'$, D outputs 1. Otherwise, D outputs 0.

If D guesses i^*, j^*, t_1, and t_2 correctly, then D exactly simulates Game_0 or Game_1 depending on whether $\tau \in \{\tau_t, \tau_r\}$ is a real session key of KE_W. So, the probability of success of D

http://dx.doi.org/10.4218/etrij.14.0113.1071

depends on whether D guesses i', j', t_1, and t_2 correctly as follows:

$$\begin{aligned}\mathrm{Adv}_{D,\mathrm{KE_W}} &= \Pr[D()=1|\tau \leftarrow sk_{\mathrm{KE_W}}] - \Pr[D()=1|\tau \leftarrow \{0,1\}^{\theta}] \\ &= 1/(Nq_s)^2 \times (\Pr_A[b=b'|sk=\mathrm{F}sk_{\mathrm{KE_W}}(\hbar(sid))] \\ &\quad - \Pr_A[b=b'|sk=\mathrm{F}_R(\hbar(sid)); R \leftarrow \{0,1\}^{\theta}]) \\ &= 1/2(Nq_s)^2 \times (\mathrm{Adv}^{\mathrm{Game}_0}{}_A - \mathrm{Adv}^{\mathrm{Game}_1}{}_A). \quad \blacksquare\end{aligned}$$

Proof of Claim 2.2. If the difference of advanatages of adversary A between Game_1 and Game_2 is non-negligible, then we can construct an algorithm E that breaks the pseudorandomness of F with a non-negligible probability using A. Consider a distinguisher E to break the pseudorandomness of a pseudorandom function family F. Distinguisher E is given an oracle function $f(\cdot)$ in the experiment of pseudorandomness of the function family F. It then uses $f(\cdot)$ to generate a session key for the test oracle. A more concrete description of E is as follows:

1) Distinguisher E is given an oracle function $f(\cdot)$. E simulates Game_1.
2) For each oracle query of A, E answers it as in Game_1 except in the case of Test(i, k) query whereby E flips a coin b. If b is equal to 1, E returns $f(\hbar(sid_{i,k}))$. Otherwise, E returns a random value selected from $\{0,1\}^{\theta}$.
3) Assume that A outputs b' then and quits. If Test(i,k) was queried but there exists no oracle $\Pi_{j,k'}$ such that $\mathrm{strID}_{j,k'} = \mathrm{strID}_{i,k}$ and $\mathrm{strTIME}_{j,k'} \neq \mathrm{strTIME}_{i,k}$, then E fails and stops. If $b=b'$, E outputs 1. Otherwise then E outputs 0.

Distinguisher E simulates Game_1 or Game_2 depending on whether $f(\cdot)$ is a function from F. So, the following inequality holds:

$$\begin{aligned}\mathrm{Adv}^{\mathrm{PRF}}{}_E &= \Pr[E^{f(\cdot)}=1|K \leftarrow \{0,1\}^{\theta}; f=\mathrm{F}_K] \\ &\quad - \Pr[E^{f(\cdot)}=1|h \leftarrow \mathrm{Rand}^{\{0,1\}^{\theta}\rightarrow\{0,1\}^{\theta}}; f=h] \\ &= \Pr_A[b=b'|K \leftarrow \{0,1\}^{\theta}; f=\mathrm{F}_K] \\ &\quad - \Pr_A[b=b'|h \leftarrow \mathrm{Rand}^{\{0,1\}^{\theta}\rightarrow\{0,1\}^{\theta}}; f=h] \\ &= 1/2\,(\mathrm{Adv}^{\mathrm{Game}_1}{}_A - \mathrm{Adv}^{\mathrm{Game}_2}{}_A). \quad \blacksquare\end{aligned}$$

If we convert HMQV in the KR model into $\mathrm{HMQV_S}$ in the BR model, then the converted $\mathrm{HMQV_S}$ protocol in the BR model is as follows:

	P_i (initiator)	P_j (responder)
Round 1	g^{α_i}	
Round 2		g^{α_j}

A session identifier is $sid = g^{\alpha_i} \| g^{\alpha_j}$, and a session key is $sk = \mathrm{F}sk_{\mathrm{HMQV}}(\hbar(sid))$, where $d = H(g^{\alpha_i} \| y_j)$, $e = H(g^{\alpha_j} \| y_j)$, and $sk_{\mathrm{HMQV}} = H(g^{(\alpha_i + dx_i)(\alpha_j + ex_j)})$.

Corollary 2. If F is a secure pseudorandom function, then $\mathrm{HMQV_S}$ is w-FS/KCI/SSR-secure in the BR model.

Proof of Corollary 2. From Theorem 1 and Theorem 6, Corollary 2 follows. $\blacksquare$

If we convert KAM in the JKL model into $\mathrm{KAM_S}$ in the BR model, then the converted $\mathrm{KAM_S}$ protocol in the BR model is as follows:

	P_i (initiator)	P_j (responder)
Round 1	g^{α_i}	
Round 2		g^{α_j}
Round 3	τ_i	
Round 4		τ_j

A session identifier is $sid = g^{\alpha_i} \| g^{\alpha_j} \| \tau_i \| \tau_j$, and a session key is $sk = \mathrm{F}sk_{\mathrm{KAM}}(\hbar(sid))$, where $sk_{\mathrm{KAM}} = H(g^{x_i x_j}) \oplus H(g^{\alpha_i \alpha_j})$.

Corollary 3. If F is a secure pseudorandom function, then $\mathrm{KAM_S}$ is s-FS/KCI/SSR-secure without random oracles in the BR model.

Proof of Corollary 3. From Theorem 3, Theorem 4, and Theorem 6, Corollary 3 follows. $\blacksquare$

Table 1 shows the comparison of complexities between the original schemes and the converted schemes.

Table 1. Detailed comparison of original and converted schemes.

	Communication complexity	Computational complexity	Security model
HMQV [2]	2 rounds	3.5 exp.†	KR model
KAM [22]	2 rounds	6 exp.	JKL model
$\mathrm{HMQV_S}$	2 rounds	3.5 exp. + 1 PRF	BR model
$\mathrm{KAM_S}$	4 rounds	6 exp. + 1 PRF	BR model

†The basic HMQV requires 2.5 exponentiations. But, the group membership tests of static and ephemeral Diffie–Hellman messages are required for SSR security, as noted in [2].

VI. Conclusion

We have shown that a key-establishment protocol secure in a specific security model might not be secure in a more restricted security model. For example, we have shown that HMQV is not secure in the Bellare–Rogaway security model. We have suggested compilers by which we can convert a key-establishment protocol secure in a specific security model into a key-establishment protocol secure in a more restricted security model.

References

[1] M. Bellare and P. Rogaway, "Entity Authentication and Key

Distribution," *CRYPTO*, Santa Barbara, CA, USA, vol. 773, Aug. 22–26, 1994, pp. 232–249.

[2] H. Krawczyk, "HMQV: A High-Performance Secure Diffie–Hellman Protocol," *CRYPTO*, Santa Barbara, CA, USA, vol. 3621, Aug. 14–18, 2005, pp. 546–566.

[3] I.R. Jeong, J. Katz, and D.H. Lee, "One-Round Protocols for Two-Party Authenticated Key Exchange," *ACNS*, Yellow Mountain, China, vol. 3089, June 8–11, 2004, pp. 220–232.

[4] W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, Nov. 1976, pp. 644–654.

[5] D. Denning and G. Sacco, "Timestamps in Key Distribution Protocols," *Commun. ACM,* vol. 24, no. 8, Aug. 1981, pp. 533–536.

[6] S. Blake-Wilson and A. Menezes, "Authenticated Diffie–Hellman Key Agreement Protocols," *SAC*, Kingston, Ontario, Canada, Aug. 17–18, 1998, pp. 339–361.

[7] L. Law et al., "An Efficient Protocol for Authenticated Key Agreement," *Des. Codes Cryptography*, vol. 28, no. 2, Mar. 2003, pp. 119–134.

[8] A. Menezes, M. Qu, and S. Vanstone, "Some New Key Agreement Protocols Providing Mutual Implicit Authentication," *SAC*, Ottawa, Ontario, Canada, May 18–19, 1995, pp. 22–32.

[9] R. Canetti and H. Krawczyk, "Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels," *EUROCRYPT*, Innsbruck, Austria, May 6–10, 2001, pp. 453–474.

[10] W. Diffie, P. Oorschot, and M. Wiener, "Authentication and Authenticated Key Exchanges," *Des. Codes, Cryptography*, vol. 2, no. 2, June 1992, pp. 107–125.

[11] M. Bellare, R. Canetti, and H. Krawczyk, "A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols," *STOC*, Dallas, TX, USA, May 23–26, 1998, pp. 419–428.

[12] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key Agreement Protocols and their Security Analysis," *IMA Int. Conf. Cryptography Coding*, vol. 1355, Cirencester, UK, Dec. 17–19, 1997, pp. 30–45.

[13] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," *EUROCRYPT*, Amsterdam, Netherlands, vol. 2332, Apr. 28–May 2, pp. 337–351.

[14] American National Standard (ANSI) X9.42-2001, *Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*, 2001.

[15] IEEE 1363-2000, *Standard Specifications for Public Key Cryptography*, 2000.

[16] ISO/IEC IS 15946-3, *Information Technology - Security Techniques Cryptographic Techniques Based on Elliptic Curves – Part 3: Key Establishment*, 2002.

[17] NIST Special Publication 800-56 (DRAFT), *Recommendation on Key Establishment Schemes*, 2003.

[18] J. Stasak, "*NSAs Elliptic Curve Licensing Agreement*," presentation to the IETF's Security Area Advisory Group, 2004. https://www.ietf.org/proceedings/61/slides/saag-2/saag3.ppt

[19] K. Choo, C. Boyd, and Y. Hitchcock, "Examining Indistinguishability-Based Proof Models for Key Establishment Protocols," *ASIACRYPT*, Chennai, India, vol. 3788, Dec. 4–8, 2005, pp. 585–604.

[20] T. Okamoto, "Authenticated Key Exchange and Key Encapsulation in the Standard Model," *ASIACRYPT*, Kuching, Sarawak, Malaysia, vol. 4833, Dec. 2–6, 2007, pp. 474–484.

[21] J. Katz and M. Yung, "Scalable Protocols for Authenticated Group Key Exchange," *CRYPTO*, Santa Barbara, CA, USA, Aug. 17–21, 2003, pp. 110–125.

[22] I.R. Jeong, J.O. Kwon, and D.H. Lee, "A Diffie–Hellman Key Exchange Protocol without Random Oracles," *CANS*, Suzhou, China, vol. 4301, Dec. 8–10, 2006, pp. 37–54.

[23] I.R. Jeong and D.H. Lee, "Key Agreement for Key Hypergraph," *Comput. Sec*., vol. 26, no. 7–8, Dec. 2007, pp. 452–458.

[24] I.R. Jeong and D.H. Lee, "Parallel Key Exchange," *J. Univ. Comput. Sci*., vol. 14, no. 3, 2008, pp. 377–396.

[25] A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, Boca Raton, USA: CRC Press, 1996, pp. 490–497.

[26] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated Key Exchange Secure against Dictionary Attacks," *EUROCRYPT*, Bruges, Belgium, vol. 1807, May 14–18, 2000, pp. 139–155.

Jeong Ok Kwon received her BS degree in computer science from Dongduk Women's University, Seoul, Rep. of Korea, in 2000. She received her MS and PhD degrees in information security from Korea University, Seoul, Rep. of Korea, in 2003 and 2007, respectively. In 2007, she began work at Korea University as a post doctor. Then in 2008 she became a research professor; a position that she held until 2009. Currently, she is a senior engineer working for Integrated Security Consulting Group, Samsung SDS, Seoul, Rep. of Korea. Her current research interests include cryptography and information security.

Ik Rae Jeong received his BS and MS degrees in computer science and his PhD degree in information security from Korea University, Seoul, Rep. of Korea, in 1998, 2000, and 2004, respectively. From 2006 to 2008, he was a senior engineer at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. Currently, he is a faculty at the Graduate School of Information Security, Korea University, Seoul, Rep. of Korea. His current research areas include cryptography and theoretical computer science.