On Session Key Construction in Provably-Secure Key Establishment Protocols*

Kim-Kwang Raymond Choo, Colin Boyd, and Yvonne Hitchcock

Information Security Institute
Queensland University of Technology
GPO Box 2434, Brisbane, QLD 4001, Australia
{k.choo,c.boyd,y.hitchcock}@qut.edu.au

Abstract. We examine the role of session key construction in provably-secure key establishment protocols. We revisit an ID-based key establishment protocol due to Chen & Kudla (2003) and an ID-based protocol 2P-IDAKA due to McCullagh & Barreto (2005). Both protocols carry proofs of security in a weaker variant of the Bellare & Rogaway (1993) model where the adversary is not allowed to make any Reveal query. We advocate the importance of such a (Reveal) query as it captures the known-key security requirement. We then demonstrate that a small change to the way that session keys are constructed in both protocols results in these protocols being secure without restricting the adversary from asking the Reveal queries in most situations. We point out some errors in the existing proof for protocol 2P-IDAKA, and provide proof sketches for the improved Chen & Kudla's protocol. We conclude with a brief discussion on ways to construct session keys in key establishment protocols.

Keywords. Key establishment protocols, provable security

1 Introduction

Key establishment protocols are used for distributing shared keying material in a secure manner. For example, today's cryptosystems, such as AES, use key establishment schemes to establish shared keying material. However, despite their importance, the difficulties of obtaining a high level of assurance in the security of almost any new, or even existing, protocols are well illustrated with examples of errors found in many such protocols years after they were published.

The treatment of computational complexity analysis 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 [3] who provided the first formal definition for a model of adversary capabilities with an associated definition of security (which we refer to as the

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

BR93 model in this paper). Since then, many research efforts have been oriented towards this end which have resulted in numerous protocols with accompanying computational proofs of security proposed in the literature. In 1995, Bellare and Rogaway analysed a three-party server-based key distribution (3PKD) protocol [4] using an extension to the BR93 model. A more recent revision to the BR93 model was proposed in 2000 by Bellare, Pointcheval and Rogaway [2]. In independent yet related work, Bellare, Canetti, & Krawczyk [1] 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 [8], and will be referred to as the CK2001 model in this paper.

Protocols in the BR93 Model. The BR93 model is probably one of the most widely used proof models in the computational complexity approach for protocol analysis. In the model, the probabilistic polynomial-time (PPT) adversary controls all the communications that take place between parties via a pre-defined set of oracle queries, namely: Send, Reveal, and Corrupt. The Reveal query allows an adversary to expose session keys for uncorrupted parties, whilst the Corrupt query allows the adversary to corrupt any principal at will, and thereby learn the complete internal state of the corrupted principal. We observe that several protocols proven secure in the BR93 model restrict the adversary from asking the Reveal query. However, we argue that such a query is realistic in a real-world implementation as an adversary is often assumed to have the capability to acquire session keys. Such a (Reveal) query is essential as it allows us to model the scenario whereby each session key generated in one protocol round is independent and determines whether the particular session key will be exposed if other secret keys are compromised. In other words, the Reveal query captures the known-key security requirement in key establishment protocols, whereby a protocol should still achieve its goal in the face of a malicious adversary who has learned some other session keys [6, 11]. In addition, omission of the Reveal query to the owner of the Test session in the proof model could also result in protocols vulnerable to reflection attacks being proven secure in such a model.

Case Studies. We revisit an ID-based key establishment protocol due to Chen & Kudla [9] and an ID-based protocol 2P-IDAKA due to McCullagh & Barreto [15]. Both protocols are role-symmetric and carry proofs of security in the BR93 model. However, the existing proofs of both protocols restrict the adversary from asking any Reveal query. Their arguments follow on from earlier work of Blake-Wilson, Johnson, & Menezes [5] who pointed out that it does not seem possible for role-symmetric protocols to be secure in the BR93 model if the Reveal query is allowed. In recent work, Jeong, Katz, & Lee [12] present two protocols $\mathcal{TS}1$ and $\mathcal{TS}2$, both with proofs of security in the BR93 model. This work contradicts the claim of Blake-Wilson *et al.* [5] as both protocols $\mathcal{TS}1$ and $\mathcal{TS}2$ are similar to the protocols analysed by Blake-Wilson *et al.* [5] in the Bellare & Rogaway (1995) model [4], but without restricting the adversary from asking the Reveal query.

We examine the existing arguments on the restriction of the Reveal query. We then demonstrate that by making a simple change to the construction of the session key (and not changing the protocol details), we are able to prove Chen & Kudla's protocol secure in an intermediate variant of the BR93 model whereby the adversary, $\mathcal{A}$, is allowed to ask all the queries available in the model except asking Reveal queries to the sessions owned by the partner of the target Test session. Although we are unable to prove the improved protocol secure in the BR93 model without restricting $\mathcal{A}$ from asking the Reveal query due to some technicality, the improved protocol does not appear to be suffering from any insecurities even if we allow $\mathcal{A}$ to ask any Reveal queries to the perceived partner of the target Test session. Furthermore, by allowing $\mathcal{A}$ to ask Reveal queries directed at the owner of the Test session in our proof, effectively means that the improved Chen & Kudla's protocol is secure against reflection attacks. We reveal some errors in the existing proof of protocol 2P-IDAKA [15] as well as the observation that the proof is in a restricted BR93 model whereby $\mathcal{A}$ does not generate the input to the Test session, which is not a normal assumption in the Bellare–Rogaway models [2–4].

The Importance of Session Key Construction. We observe that there is neither a formal definition of session key construction in the proof models nor the existence of a rule of thumb on how session keys in key establishment protocols should be constructed. Our case studies illustrate that the way session keys are constructed can have an impact on the security of the protocol in the model. It appears that certain ways of constructing a session key may contribute to the security of a key establishment protocol.

Surprisingly, no one has pointed out the importance of session key construction despite its significance to the security of key establishment protocols. Of course, we do not claim that session keys constructed in our proposed fashion will necessarily result in a provably-secure protocol as the security of the protocol is based on many other factors, such as the underlying cryptographic primitives used. However, we do claim that having a sound construction of session keys will reduce the number of possible attacks on the key establishment protocol.

Main Contributions. We regard the main contributions of this paper to be threefold:

1. demonstrating that the ID-based protocols of Chen & Kudla and McCullagh & Barreto can be proven secure in an intermediate BR93 model whereby the restriction of the Reveal query is only on the responder partner and the owner of the Test session respectively,
2. identifying the importance of session key constructions in key establishment protocols and contributing towards a better understanding of how to construct secure session keys in key establishment protocols, and
3. identifying errors in the existing proof of protocol 2P-IDAKA [15].

Organization. Section 2 provides an informal overview of the BR93 model. Section 3 revisits the Chen–Kudla ID-based key establishment protocol. We present the arguments of the existing proof on why the Reveal query is not allowed, and present an improved protocol. We then explain why the Reveal query cannot be answered if the adversary $\mathcal{A}$ ask any Reveal queries to the partner player of the target Test session. We conclude this section with a sketch of the proof for the improved protocol. Section 4 revisits the McCullagh–Barreto protocol 2P-IDAKA. Similarly to Section 3, we present the arguments of the existing proof on why the Reveal query is not allowed. We also identify some errors in the existing proof of the protocol. We then present an improved protocol. Section 5 presents our proposal on how session keys should be constructed. Section 6 presents the conclusions.

2 The BR93 Model

In this section, a brief overview of the BR93 model is provided primarily for the benefit of the reader in understanding the model [3].

2.1 Adversarial Powers

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 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 and $\Psi^j_{U_1,U_2}$ is defined to be the j^{th} 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 as follows:

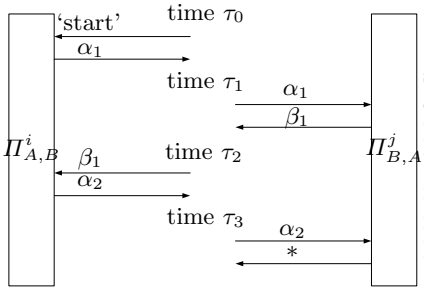
- $\mathsf{Send}(U_1, U_2, i, m)$ query computes a response according to the protocol specification and decision on whether to accept or reject yet, and returns them to $\mathcal{A}$.
- The client oracle, $\Pi^i_{U_1,U_2}$, upon receiving a $\mathsf{Reveal}(U_1, U_2, i)$ query, and if it has accepted 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. Note that such a query does not exist in the original BR93 model, but generally added by those using this model. In the Bellare & Rogaway (1995) model [4], 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.

2.2 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 [3]) *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 $\Pi^i_{A,B}$ 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*

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



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.

Fig. 1. Matching conversation [3]

2.3 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.*

2.4 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 $\mathcal{G}$, whose setting is explained in Table 1.

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 Corrupt and/or Session-Key Reveal and/or Session-State Reveal queries (depending on the individual proof model) 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.

Table 1. Setting of game $\mathcal{G}$

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 $\mathsf{Adv}^{\mathcal{A}}(\mathsf{k}) = 2 \times \mathsf{Pr}[\mathsf{b} = \mathsf{b}'] - 1$. Definition 3 describes security for the BR93 model.

Definition 3 (BR93 Definition of Security [3]) *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 3 are satisfied, then $\Pi^i_{A,B}$ and $\Pi^j_{B,A}$ will also have the same session key.

3 Chen–Kudla ID-Based Authenticated Key Establishment Protocol

Figure 2 describes protocol 2 of Chen–Kudla. There are two entities in the protocols, namely initiator, A, and responder, B. The notation used in the protocols is as follows: $S_A = sQ_A$ and $S_B = sQ_B$ denote the private keys of A and B respectively, $\mathcal{H}$ denotes some secure hash function, $Q_A = \mathcal{H}(ID_A)$, $Q_A = \mathcal{H}(ID_B)$, $W_A = aQ_A$ and $W_B = bQ_B$ where W_A and W_B denote the ephemeral public keys of A and B respectively, and a and b are the ephemeral private keys of A and B respectively. At the end of the protocol execution, both A and B accept the session key $SK_{AB} = \hat{e}(S_A, W_B + aQ_B)$ and $SK_{BA} = \hat{e}(W_A + bQ_A, S_B)$ respectively, where $SK_{AB} = SK_{BA} = \hat{e}(Q_A, Q_B)^{s(a+b)}$.

A		B
$a \in_R \mathbb{Z}_q^*$	$\xrightarrow{W_A = aQ_A}$	$b \in_R \mathbb{Z}_q^*$
$K_{AB} = \hat{e}(S_A, W_B + aQ_B)$	$\xleftarrow{W_B = bQ_B}$	$K_{BA} = \hat{e}(W_A + bQ_A, S_B)$
$K_{AB} = K_{BA} = \hat{e}(Q_A, Q_B)^{s(a+b)}$		
$SK_{AB} = \mathcal{H}(K_{AB}) = SK_{BA} = \mathcal{H}(K_{BA})$		

Fig. 2. Chen–Kudla Protocol 2

3.1 Existing Arguments on the Restriction of Reveal Query

In the existing proof by Chen & Kudla [9, Proof of Theorem 1], they indicated that no Reveal query is allowed due to the description provided in Figure 3, where Figure 3 describes the execution of the protocol in the presence of a malicious adversary, $\mathcal{A}$.

A		$\mathcal{A}$		B
$a \in_R \mathbb{Z}_q^*$	$\xrightarrow{W_A = aQ_A}$	Intercept		
		$c \in_R \mathbb{Z}_q^*$	$\xrightarrow{W_A + cQ_A}$	
	$\xleftarrow{W_B + cQ_B}$	Intercept	$\xleftarrow{W_B = bQ_B}$	$b \in_R \mathbb{Z}_q^*$
$K_{AB} = \hat{e}(S_A, W_B + cQ_B + aQ_B)$			$K_{BA} = \hat{e}(W_A + bQ_A + cQ_A, S_B)$	
$K_{AB} = K_{BA} = \hat{e}(Q_A, Q_B)^{s(a+b+c)}$				
$SK_{AB} = \mathcal{H}(K_{AB}) = SK_{BA} = \mathcal{H}(K_{BA})$				

Fig. 3. Execution of Chen-Kudla protocol 2 in the presence of a malicious adversary

At the end of the protocol execution, neither A nor B are partnered since they do not have matching conversations (as described in Definition 1 in Section 2), as A's transcript is $(W_A, W_B + cQ_B)$ whilst B's transcript (as described in Section 2) is $(W_A + cQ_A, W_B)$. However, both A and B accept the same session key $K_{AB} = K_{BA} = \hat{e}(Q_A, Q_B)^{s(a+b+c)}$. Therefore, $\mathcal{A}$ is able to trivially expose a fresh session key by asking a Reveal query to a non-partner oracle. Therefore, the protocol will not be secure if $\mathcal{A}$ is allowed access to a Reveal query. Similar arguments apply for the remaining three protocols of Chen & Kudla [9].

3.2 Improved Chen–Kudla Protocol

Let A's transcript be denoted by $\mathcal{T}_A$ and B's transcript be denoted by $\mathcal{T}_B$. Consider the scenario whereby session keys of A and B (denoted as SK_{AB} and SK_{BA} respectively) are constructed as

$$\begin{aligned} SK_{AB} = \mathcal{H}(K_{AB}) &= \mathcal{H}(A||B||\mathcal{T}_A||\hat{e}(S_A, W_B + aQ_B)) \\ &= \mathcal{H}(A||B||\mathcal{T}_A||\hat{e}(Q_A, Q_B)^{s(a+b)}), \\ SK_{BA} = \mathcal{H}(K_{BA}) &= \mathcal{H}(A||B||\mathcal{T}_B||\hat{e}(W_A + bQ_A, S_B) \\ &= \mathcal{H}(A||B||\mathcal{T}_B||\hat{e}(Q_A, Q_B)^{s(a+b)}) = SK_{AB} \end{aligned}$$

instead. Evidently, the attack outlined in Figure 3 will no longer work since a non-matching conversation (i.e., $\mathcal{T}_A \neq \mathcal{T}_B$) will also mean that the session key is different, as shown below:

$$\begin{aligned} &SK_{AB} = \mathcal{H}(K_{AB}) = \mathcal{H}(A||B||aQ_A||(b+c)Q_B||\hat{e}(S_A, W_B + aQ_B)), \\ &SK_{BA} = \mathcal{H}(K_{BA}) = \mathcal{H}(A||B||(a+c)Q_A||bQ_B||\hat{e}(W_A + bQ_A, S_B)) \neq SK_{AB}. \end{aligned}$$

Similarly, a reflection attack or an unknown key share attack would not work against the protocol since the construction of the session key introduces role asymmetry and the identities of the participants. In other words, session keys will be different when the roles of the same principal switch. Therefore, the adversary, $\mathcal{A}$, appears to be unable to gain information about such fresh session key(s).

3.3 Sketch of New Proof

At first glance, it would seem that by fixing the attack outlined in Section 3.1, we have addressed the reasons why no Reveal query was allowed that was outlined in the existing proofs, and would be able to prove the improved protocol secure in the unrestricted BR93 model. However, we demonstrate that this is not possible unless we restrict the adversary from asking any Reveal queries to the partner of the Test session, as explained in Figure 4. However, by allowing the adversary

to ask Reveal queries directed at the owner of the Test session (in our proof), we effectively prove the improved protocol secure against reflection attacks.

Recall that the general notion of the proof is to assume that there exists an adversary $\mathcal{A}$ who can gain a non-negligible advantage in distinguishing the test key in the game described in Section 2.4, and use $\mathcal{A}$ to break the underlying BDH problem. In other words, we build an adversary, $\mathcal{A}_{\mathcal{BDH}}$, against the BDH problem using $\mathcal{A}$. The objective of $\mathcal{A}_{\mathcal{BDH}}$ is to compute and output the value $\widehat{e}(P,P)^{xyz} \in G_2$ when given a bilinear map $\widehat{e}$, a generator of P of G_1, and a triple of elements $xP, yP, zP \in G_1$ with $x, y, z \in \mathbb{Z}_q^*$, where q is the prime order of the distinct groups G_1 and G_2.

Let oracle $\Pi^u_{A,B}$ be the initiator associated with the target Test session, and oracle $\Pi^v_{B,A}$ be the responder partner to $\Pi^u_{A,B}$. $\mathcal{A}_{\mathcal{BDH}}$ needs to simulate all responses to queries from $\mathcal{A}$, including the random oracle, $\mathcal{H}$. The proof specifies that $\mathcal{A}_{\mathcal{BDH}}$ can create all public/private key pairs for all players, except a randomly chosen player J. Let (Q_U, S_U) denote the public/private keys of players U other than J (where $S_U = xQ_U$). $\mathcal{A}_{\mathcal{BDH}}$ is unable to compute the private key of J because $\mathcal{A}_{\mathcal{BDH}}$ is trying to solve the BDH problem, which is embedded in the public key of J.

Figure 4 shows a possible sequence of adversary actions and the responses generated by $\mathcal{A}_{\mathcal{BDH}}$. It can be seen that $\mathcal{A}$ will be able to distinguish between the simulation provided by $\mathcal{A}_{\mathcal{BDH}}$ and the actual protocol if it carries out this sequence of actions, since with overwhelming probability, $v \neq SK_{BC}$ (recall that v is randomly chosen). Hence, $\mathcal{A}_{\mathcal{BDH}}$ cannot answer any Reveal directed at the partner of the target Test session.

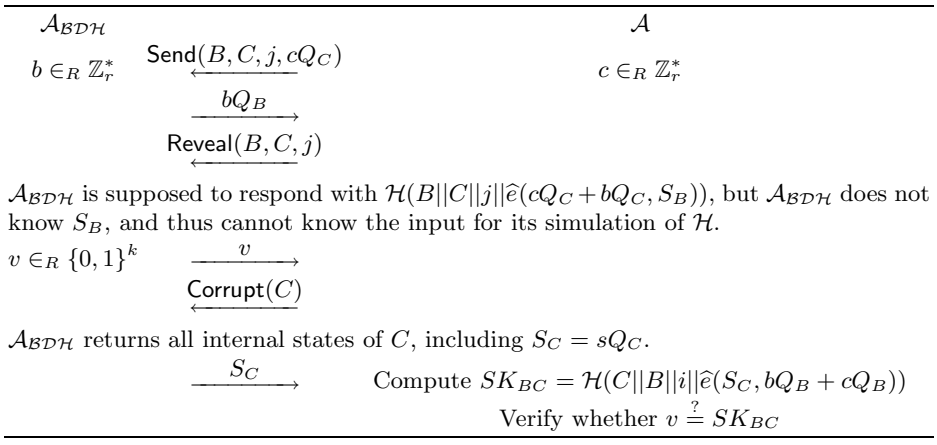


Fig. 4. An example simulation of Chen–Kudla protocol 2

Theorem 1 *The improved Chen–Kudla protocol 2 is a secure authenticated key establishment protocol in the sense of Definition 3 if the Bilinear Diffie-Hellman*

(BDH) problem is hard and the hash function, $\mathcal{H}$, is a random oracle, and the adversary $\mathcal{A}$ does not ask any Reveal *queries to any sessions owned by the partner player associated with the* Test *session.*

The proof of Theorem 1 generally follows that of Chen & Kudla [9, Proof of Theorem 1], except that we allow $\mathcal{A}$ to ask Reveal queries (but not to the partner player of the Test session). The details of the game simulation remain unchanged to that presented by Chen & Kudla [9, Proof of Theorem 1], except that we allow $\mathcal{A}$ to ask Reveal queries (but not to the partner player of the Test session), as given in Figure 5.

Queries	Actions
$\mathsf{Send}(U_1, U_2, i)$	$\mathcal{A}_{\mathcal{BDH}}$ answers all Send queries in the same fashion as the proof simulation presented by Chen & Kudla.
$\mathsf{Corrupt}(U, K)$	$\mathcal{A}_{\mathcal{BDH}}$ answers all Corrupt queries in the same fashion as the proof simulation presented by Chen & Kudla.
$\mathsf{Test}(U_1, U_2, i)$	$\mathcal{A}_{\mathcal{BDH}}$ answers the Test query in the same fashion as the proof simulation presented by Chen & Kudla.
$\mathcal{H}(U_1 \| U_2 \| i \| te(m))$	$\mathcal{A}_{\mathcal{BDH}}$ will return a random value, $v \in_R \{0,1\}^k$ where k is the security parameter and store m in a list of tuples.
$\mathsf{Reveal}(U_1, U_2, i)$	If oracle $\Pi^i_{U_1,U_2}$ is not an oracle associated with the test session (or partner of such an oracle), and U_1 is not player J where $\mathcal{A}_{\mathcal{BDH}}$ did not generate the contents of the Send query to $\Pi^i_{U_1,U_2}$, then $\mathcal{A}_{\mathcal{BDH}}$ returns the associated session key. Otherwise $\mathcal{A}_{\mathcal{BDH}}$ terminates and halts the simulation. We observe that if $\mathcal{A}_{\mathcal{BDH}}$ halts because $U_1 = J$, the Test session chosen by $\mathcal{A}$ must be different to that desired by $\mathcal{A}_{\mathcal{BDH}}$, so even if the simulation had not halted here, it would have halted later.

Fig. 5. $\mathcal{A}_{\mathcal{BDH}}$ simulates the view of $\mathcal{A}$ by answering all Send, Reveal, Corrupt, and Test oracle queries of $\mathcal{A}$.

Hence, $\mathcal{A}_{\mathcal{BDH}}$ is able to simulate the view of $\mathcal{A}$ perfectly by answering all oracle queries of $\mathcal{A}$ as specified in Figure 5. Upon the conclusion of the game (i.e., $\mathcal{A}$ is done), $\mathcal{A}_{\mathcal{BDH}}$ chooses a random element in the list of tuples and outputs it. The probability that $\mathcal{A}_{\mathcal{BDH}}$ did not abort at some stage and produces the correct output remains non-negligible. This concludes the sketch of the proof of the theorem.

4 2P-IDAKA Protocol

In recent work, McCullagh & Barreto [15] proposed a two-party ID-based authenticated key agreement (2P-IDAKA) protocol with a proof of security in a weaker variant of the BR93 model whereby the adversary is not allowed to

ask Reveal queries. Figure 6 describes the 2P-IDAKA protocol. There are two entities in the protocol, namely an initiator player A and a responder player B. Notation used in the protocols is as follows: $(s+a)P$ denotes the public key of A, $A_{pri} = ((s+a))^{-1}P$ denotes the private key of A, $(s+b)P$ denotes the public key of B, and $B_{pri} = ((s+b))^{-1}P$ denotes the private key of B. At the end of the protocol execution, both A and B accept session keys $SK_{AB} = \hat{e}(B_{KA}, A_{pri})^{x_a} = \hat{e}(P,P)^{x_a x_b} = SK_{BA}$.

A		B
$x_a \in_R Z_r^*$	$A_{KA} = x_a(s+b)P \xrightarrow{\quad}$	$x_b \in_R Z_r^*$
$\hat{e}(B_{KA}, A_{pri})^{x_a} = \hat{e}(P,P)^{x_a x_b}$	$B_{KA} = x_b(s+a)P \xleftarrow{\quad}$	$\hat{e}(A_{KA}, B_{pri})^{x_b} = \hat{e}(P,P)^{x_a x_b}$

Fig. 6. McCullagh–Barreto 2P-IDAKA protocol

4.1 Why Reveal Query is Restricted

No Reveal query is allowed on the 2P-IDAKA protocol [10] due to the description provided in Figure 7.

A		$\mathcal{A}$		B
$x_a \in_R \mathbb{Z}_r^*$	$A_{KA} = x_a(s+b)P \xrightarrow{\quad}$	Intercept		
		$x_E \in_E Z_r^*$		
		Impersonate A	$\xrightarrow{A_{KA} \cdot x_E}$	$x_b \in_R \mathbb{Z}_r^*$
		Intercept	$B_{KA} = x_b(s+a)P \xleftarrow{\quad}$	
	$\xleftarrow{B_{KA} \cdot x_E}$	Impersonate B		
$SK_A = \hat{e}(x_b(s+a)P \cdot x_E, A_{pri})^{x_a} = \hat{e}(P,P)^{x_a x_b x_E} = SK_B$				

Fig. 7. Execution of 2P-IDAKA protocol in the presence of a malicious adversary

In the protocol execution shown in Figure 7, both A and B have accepted the same session key (i.e., $SK_A = SK_B$). However, both A and B are non-partners since they do not have matching conversations as A's transcript is $(A_{KA}, B_{KA} \cdot x_E)$ whilst B's transcript is $(A_{KA} \cdot x_E, B_{KA})$. By sending a Reveal query to either A or B, $\mathcal{A}$ is able to trivially expose a fresh session key by asking a Reveal query to either A or B. Hence, the 2P-IDAKA protocol shown in Figure 6 is not secure since $\mathcal{A}$ is able to obtain the session key of a fresh oracle of a non-partner oracle by revealing a non-partner oracle holding the same key, in violation of the key establishment goal.

4.2 Errors in Existing Proof

The general notion of the existing proof of McCullagh & Barreto [15, Proof of Theorem 1], to assume that there exists an adversary $\mathcal{A}$ who can gain a non-negligible advantage in distinguishing the test key in the game described in Section 2.4, and use $\mathcal{A}$ to break the underlying Bilinear Inverse Diffie–Hellman Problem (BIDHP). In other words, an adversary, $\mathcal{A}_{\mathcal{BIDHP}}$, against the BIDHP is constructed using $\mathcal{A}$. The objective of $\mathcal{A}_{\mathcal{BIDHP}}$ is to compute and output the value $\widehat{e}(P,P)^{\alpha^{-1}\beta}$ when given $P, \alpha P, \beta P$ for $x, y, z \in \mathbb{Z}_r^*$.

Error 1: In the existing proof, the public and private key pairs for some player, U_i, are selected as $((u-s)P, u^{-1}P)$, in contradiction to their description in the protocols where $((s+u)P, (s+u)^{-1}P)$ is given instead. The adversary, $\mathcal{A}$, is then able to tell that the public and private key pairs do not match by simply corrupting any player, as shown in Figure 8.

$\mathcal{A}_{\mathcal{BIDHP}}$		$\mathcal{A}$
Return all internal state of U,	$\xleftarrow{\mathsf{Corrupt}(U)}$	
including $(u)^{-1}P$	$\xrightarrow{u^{-1}P}$	Compute $\widehat{e}(uP - sP, (u)^{-1}P)$

Fig. 8. Illustration of error 1

We can check whether a public and private key pair match by computing $\widehat{e}((s+u)P, (s+u)^{-1}P) = \widehat{e}((P,P)^{(s+u)(s+u)^{-1}} = \widehat{e}(P,P)$. However, as outlined in Figure 8, when $\mathcal{A}$ computes the public and private key pair of U, $\widehat{e}(uP - sP, (u)^{-1}P) = \widehat{e}((u-s)P, u^{-1}P) = \widehat{e}(P,P)^{(u-s)u^{-1}} = \widehat{e}(P,P)^{1-su^{-1}} \neq \widehat{e}(P,P)$. $\mathcal{A}$ trivially knows that the public and private key pairs of U do not match. Hence, the existing proof is invalidated.

Error 2: We observed that the parameter $\beta P = x_j \alpha P$ given in the existing proof should be $\beta P = x_j(y_i - s)P$ instead, as explained in Figure 9. In Figure 9, we assume that error 1 has been fixed. The public/private key pair of I (the partner player associated with the Test session is $((y_i - s)P, (y_i - s)^{-1}P)$, the public key of J (the owner of the Test session) is αP, and the private key of J (i.e., $\alpha^{-1}P$)) is unknown to both $\mathcal{A}_{\mathcal{BIDHP}}$ and $\mathcal{A}$.

It is obvious from Figure 9 that we cannot have the values of both $x_i P$ and $x_j P$ computed using the public key of J, αP (at least one of $x_i P$ and $x_j P$ have to be computed using the public key of I). To check, we compute $\widehat{e}(P,P)^{x_t x_j} = \widehat{e}(P,P)^{x_i \alpha^{-1} \beta \alpha^{-1}} \neq \widehat{e}(P,P)^{\alpha^{-1}\beta}$, which is what $\mathcal{A}_{\mathcal{BIDHP}}$ is trying to solve. Hence, the correct value for $\beta P = x_j \alpha P$ given in the existing proof should be $\beta P = x_j(y_i - s)P$ instead.

$\mathcal{A}_{\mathcal{BIDHP}}$		$\mathcal{A}$
$x_i \in_R \mathbb{Z}_r^*$	$\xleftarrow{\mathsf{Send}(J, I, i, ' start')}$	
	$\xrightarrow{x_i P}$	$\mathcal{A}$ is given $\beta P = x_j \alpha P$ as input from j
$x_i P = x_t(\alpha P)$	$\xleftarrow{\mathsf{Send}(I, J, j, \beta P)}$	

Fig. 9. Illustration of error 2

Further remarks: We observe that for the existing proof to work, we would have to assume that the inputs to the Test session originated with the simulator, $\mathcal{A}_{\mathcal{BIDHP}}$, and not the adversary, $\mathcal{A}$. However, this is not a normal assumption and resricts the BR93 model. In fact, if a slightly different assumption were made in the proof of the improved Chen & Kudla's protocol in Section 3.3, namely that if B is the partner of the Test session, then all Send query inputs to sessions of B that are later revealed were generated by $\mathcal{A}_{\mathcal{BDH}}$, then the proof in Section 3.3 would not have to restrict Reveal queries to B.

Consequences of errors in security proofs: Protocol implementers (usually non-specialists and/or industrial practitioners) will usually plug-and-use existing provably-secure protocols without reading the formal proofs of the protocols [13]. Errors in security proofs or specifications themselves certainly will certainly undermine the credibility and trustworthiness of provably-secure protocols in the real world.

4.3 Improved 2P-IDAKA Protocol

Let A's transcript be denoted by $\mathcal{T}_A$ and B's transcript be denoted by $\mathcal{T}_B$. Consider the scenario whereby session keys of A and B are constructed as

$$SK_{AB} = \mathcal{H}(A||B||\mathcal{T}_A||\hat{e}(B_{KA}, A_{pri})^{x_a}) = \mathcal{H}(A||B||\mathcal{T}_A||\hat{e}(P,P)^{x_a x_b}),$$
$$SK_{BA} = \mathcal{H}(A||B||\mathcal{T}_B||\hat{e}(A_{KA}, B_{pri})^{x_b}) = \mathcal{H}(A||B||\mathcal{T}_B||\hat{e}(P,P)^{x_a x_b}) = SK_{AB}$$

instead. Evidently, the attack outlined in Figure 7 will no longer be valid since a non-matching conversation (i.e., $\mathcal{T}_A \neq \mathcal{T}_B$) will also mean that the session key is different, as shown below:

$$SK_{AB} = \mathcal{H}(A||B||x_a(s+b)P||(x_b \cdot x_E)(s+a)P||\hat{e}(B_{KA}, A_{pri})^{x_a}),$$
$$SK_{BA} = \mathcal{H}(A||B||(x_a \cdot x_E)(s+b)P||x_b(s+a)P||\hat{e}(A_{KA}, B_{pri})^{x_b}) \neq SK_{AB}.$$

Therefore, $\mathcal{A}$ is unable to gain information about any fresh session key(s).

Figure 10 illustrates why Reveal queries directed at the owner of the Test session cannot be answered by $\mathcal{A}_{\mathcal{BDH}}$. Note that $\Pi^j_{J,C}$ is not the target Test session.

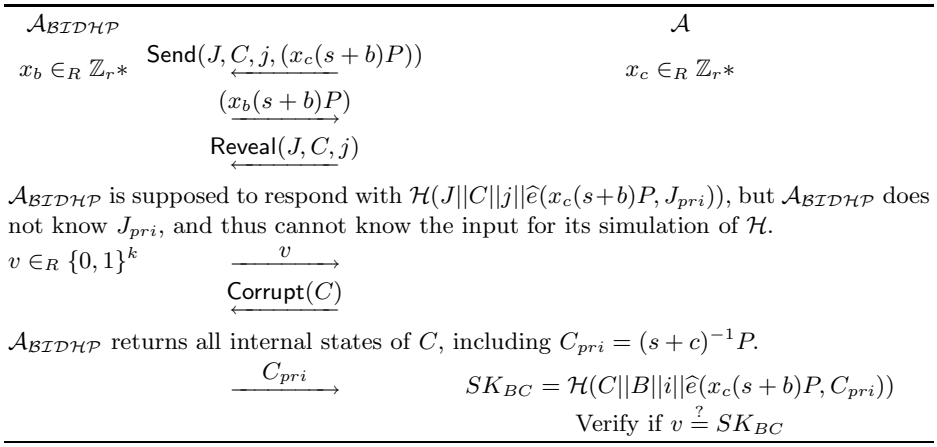


Fig. 10. An example simulation of McCullagh–Barreto 2P-IDAKA protocol

From Figure 10, it can be seen that $\mathcal{A}$ will be able to distinguish between the simulation provided by $\mathcal{A}_{\mathcal{BIDHP}}$ and the actual protocol if it carries out this sequence of actions, since with overwhelming probability, $v \neq SK_{BC}$ (recall that v is randomly chosen). Hence, $\mathcal{A}_{\mathcal{BIDHP}}$ cannot answer any Reveal directed at the owner of the target Test session, J, unless we made a similar type of assumption in the existing proof outlined in Section 4.2 that all Send query inputs to sessions of J that are later revealed were generated by $\mathcal{A}_{\mathcal{BIDHP}}$.

5 A Proposal for Session Key Construction

In this section, we present our proposal on how session keys should be constructed. Although we do not claim that session keys constructed in this fashion will result in a secure protocol (as the security of the protocol is based on many other factors, such as the underlying cryptographic primitives used), we do claim that having a sound construction of session keys may reduce the number of possible attacks on the protocol.

We propose that session keys in key establishment protocols should be constructed in the following fashion, as shown in Table 2. The inclusion of

- the identities of the participants and their roles provides resilience against unknown key share attacks and reflection attacks since the inclusion of both the identities of the participants and role asymmetry effectively ensures some sense of direction. If the role of the participants or the identities of the (perceived) partner participants change, the session keys will also be different,
- the unique session identifiers (SIDs) ensures that session keys will be fresh, and if SIDs are defined as the concatenation of messages exchanged during the protocol execution, messages altered during the transmission will result in different session keys (providing data origin authentication), and

– some other ephemeral shared secrets and/or long-term (static) shared secrets depending on individual protocols, ensures that the session key is only known to the protocol participants.

Session key input	Properties
Identities of the participants and their roles	Resilience against unknown key share attacks [7, Chapter 5.1.2] and reflection attacks [14].
Unique session identifiers (SIDs)	Freshness and data origin authentication (assuming SIDs defined to be the concatenation of exchanged messages).
Ephemeral shared secrets and/or long-term (static) shared secrets	If the identities of the (perceived) partner participants change, the session keys will also be different.

Table 2. Construction of session key in key establishment protocols

6 Conclusion

By making a small change to the way session keys are constructed in the Chen–Kudla protocol 2 and McCullagh–Barreto protocol 2P-IDAKA, we demonstrated that the existing attacks no longer work. In addition, the Chen–Kudla protocol 2 proof was improved to be less restrictive with regard to the Reveal queries allowed. We also found some errors in the McCullagh–Barreto proof, as well as observing that it is in a restricted version of the BR93 model that assumes that the adversary does not generate the input to the Test session.

As a result of our findings, we would recommend that all provably secure protocols should construct session keys using materials comprising the identities of the participants and roles, unique session identifiers (SIDs), and some other ephemeral shared secrets and/or long-term (static) shared secrets. We hope that this work contributes towards a better understanding on how to construct secure session keys in key establishment protocols.

References

1. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A Modular Approach to The Design and Analysis of Authentication and Key Exchange Protocols. In Jeffrey Vitter, editor, *30th ACM Symposium on the Theory of Computing - STOC 1998*, pages 419–428. ACM Press, 1998.
2. Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated Key Exchange Secure Against Dictionary Attacks. In Bart Preneel, editor, *Advances in Cryptology – Eurocrypt 2000*, pages 139 – 155. Springer-Verlag, 2000. Volume 1807/2000 of Lecture Notes in Computer Science.

3. Mihir Bellare and Phillip Rogaway. Entity Authentication and Key Distribution. In Douglas R. Stinson, editor, *Advances in Cryptology - Crypto 1993*, pages 110–125. Springer-Verlag, 1993. Volume 773/1993 of Lecture Notes in Computer Science.
4. Mihir Bellare and Phillip Rogaway. Provably Secure Session Key Distribution: The Three Party Case. In F. Tom Leighton and Allan Borodin, editors, *27th ACM Symposium on the Theory of Computing - STOC 1995*, pages 57–66. ACM Press, 1995.
5. Simon Blake-Wilson, Don Johnson, and Alfred Menezes. Key Agreement Protocols and their Security Analysis. In Michael Darnell, editor, *6th IMA International Conference on Cryptography and Coding*, pages 30–45. Springer-Verlag, 1997. Volume 1355/1997 of Lecture Notes in Computer Science.
6. Simon Blake-Wilson and Alfred Menezes. Security Proofs for Entity Authentication and Authenticated Key Transport Protocols Employing Asymmetric Techniques. In Bruce Christianson, Bruno Crispo, T. Mark A. Lomas, and Michael Roe, editors, *Security Protocols Workshop*, pages 137–158. Springer-Verlag, 1997. Volume 1361/1997 of Lecture Notes in Computer Science.
7. Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*. Springer-Verlag, June 2003.
8. Ran Canetti and Hugo Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels (Extended version available from `http://eprint.iacr.org/2001/040/`). In Birgit Pfitzmann, editor, *Advances in Cryptology - Eurocrypt 2001*, pages 453–474. Springer-Verlag, 2001. Volume 2045/2001 of Lecture Notes in Computer Science.
9. Liqun Chen and Caroline 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*, pages 219–233. IEEE Computer Society Press, 2003.
10. Kim-Kwang Raymond Choo. Revisit Of McCullagh–Barreto Two-Party ID-Based Authenticated Key Agreement Protocols. Cryptology ePrint Archive, Report 2004/343, 2004. `http://eprint.iacr.org/2004/343/`.
11. Dorothy E. Denning and Giovanni Maria Sacco. Timestamps in Key Distribution Protocols. *ACM Journal of Communications*, 24(8):533–536, 1981.
12. Ik Rae Jeong, Jonathan Katz, and Dong Hoon Lee. One-Round Protocols for Two-Party Authenticated Key Exchange. In Markus Jakobsson, Moti Yung, and Jianying Zhou, editors, *Applied Cryptography and Network Security - ACNS 2004*, pages 220–232. Springer-Verlag, 2004. Volume 3089/2004 of Lecture Notes in Computer Science.
13. Neal Koblitz and Alfred Menezes. Another Look at "Provable Security". Technical report CORR 2004-20, Centre for Applied Cryptographic Research, University of Waterloo, Canada, 2004.
14. Hugo Krawczyk. SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols. In Dan Boneh, editor, *Advances in Cryptology - Crypto 2003*, pages 400–425. Springer-Verlag, 2003. Volume 2729/2003 of Lecture Notes in Computer Science.
15. Noel 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 Alfred John Menezes, editor, *Cryptographers' Track at RSA Conference - CT-RSA 2005*, pages 262–274. Springer-Verlag, 2005. Volume 3376/2005 of Lecture Notes in Computer Science.