# Tripartite Key Exchange in the Canetti-Krawczyk Proof Model*

Yvonne Hitchcock, Colin Boyd and Juan Manuel González Nieto

Information Security Research Centre, Queensland University of Technology
GPO Box 2434, Brisbane Q 4001, Australia.
{y.hitchcock,c.boyd,j.gonzaleznieto}@qut.edu.au

**Abstract**

A definition of secure multi-party key exchange in the Canetti-Krawczyk proof model is proposed, followed by a proof of the security of the Joux tripartite key agreement protocol according to that definition. The Joux protocol is then combined with two authentication mechanisms to produce a variety of provably secure key agreement protocols. The properties and efficiency of the Joux based protocols thus derived are then compared with each other and other published tripartite key agreement protocols. It is concluded that the Joux protocol can be used to generate efficient yet provably secure protocols.

## 1 Introduction

A major goal of modern cryptography is to enable two or more users on an insecure (adversary controlled) network to communicate in a confidential manner and/or ensure that such communications are authentic. In order to realize this goal, symmetric key cryptographic tools are often used due to their efficiency compared to public key techniques. However, use of such tools requires the creation of a secret key (which is typically at least 100 bits long) known only to the users communicating with each other. Because of the impracticality of each possible pair of users sharing a long term secret key, public key and/or password-based techniques are used to generate such a key when it is required. An advantage of this method of key generation is to keep different sessions independent, which enables the avoiding of replay attacks (since the wrong key will have been used for the replay) and lessens the impact of key compromise (since only one session will be exposed, not all previous communications).

Although recent progress has been made on the use of formal proof models to prove the security of key exchange protocols, one area where further work is required is the use of formal proof models in conjunction with tripartite key agreement protocols. Tripartite key agreement enables three parties to exchange a key so that they can all participate in a session. It can also be used to enable two parties to communicate in the presence of a third party who may provide chairing, auditing, data recovery or escrow services [1]. In 2000, Joux [13] proposed a tripartite key exchange protocol based on pairings on an elliptic curve (such as the Weil or Tate pairing) that required only one round, but was subject to a man-in-the-middle attack due to its lack of any authentication mechanism. Al-Riyami and Paterson [1] have modified the Joux protocol in a variety of ways to overcome this problem, yet without adding to the number of rounds required by the protocol. However, only one of their protocols was accompanied by any

*This is the full version of the extended abstract that appears in *Proceedings of the 5th International Conference on Cryptology in India (INDOCRYPT 2004)*, *Lecture Notes in Computer Science*. 

sort of formal security proof, and that proof did not allow adaptive adversaries. In fact, flaws were found in preliminary versions of their protocols, demonstrating the difficulty of ensuring protocols are not flawed without the use of formal security proofs.

In this paper, we provide security proofs for Joux-based tripartite key agreement protocols that provide (implicit) key authentication. To do this, we adopt the Canetti-Krawczyk proof model [9] (hereafter referred to as the CK-model), which was based on the model of Bellare, Canetti and Krawczyk [3]. The CK-model offers the advantage of allowing modular proofs, thus allowing different components to be proven secure separately, and then joined together to produce a secure key exchange protocol. It also leads to simpler, less error-prone proofs and the ability to construct a large number of secure protocols from a much smaller number of basic secure components.

The modularity of the CK-model is gained by applying a protocol translation tool, called an *authenticator*, to protocols proven secure in a much simplified adversarial setting where authentication of the communication links is not required. The result of such an application is secure protocols in the unsimplified adversarial setting where the full capabilities of the adversary are modelled.

Unfortunately, the definition of secure key exchange provided by the original CK-model only caters for two parties, and so a modification of the definition is required to cater for tripartite key exchange. Such a modification is proposed in this paper, in conjunction with the analysis of the security and efficiency of the Joux [13] protocol. It transpires that the Joux based protocols proposed and proven secure in this paper require the same number of messages as an ordinary discrete logarithm based tripartite Diffie-Hellman protocol. However, the Joux based protocols have smaller messages and require a comparable amount of computation.

## 2 Overview of the Canetti-Krawczyk Approach

Here a description of the CK-model is given. Further details can be found in [3] and [9]. The CK-model defines protocol principals who may simultaneously run multiple local copies of a message driven protocol. Each local copy is called a session and has its own local state. Two sessions are *matching* if each session has the same session identifier and the purpose of each session is to establish a key between the particular two parties running the sessions. A session is *expired* if the session key agreed by the session has been erased from the session owner's memory. A powerful adversary attempts to break the protocol by interacting with the principals. In addition to controlling all communications between principals, the adversary is able to *corrupt* any principal, thereby learning all information in the memory of that principal (e.g. long-term keys, session states and session keys). The adversary may impersonate a corrupted principal, although the corrupted principal itself is not activated again and produces no further output or messages. The adversary may also *reveal* internal session states or agreed session keys. The adversary must be efficient in the sense of being a probabilistic polynomial time algorithm. An *unexposed* session is one such that neither it nor a matching session has had its internal state or agreed session key revealed, and if the owner of the session or a matching session is corrupted, the corruption occurred after the key had expired at the corrupted party.

**Definition 1 (Informal)** *An* AKE *protocol is called* session key (SK-) secure *if the following two conditions are met. Firstly, if two uncorrupted parties complete matching sessions, then they both accept the same key. Secondly, suppose the adversary chooses as a "test session" one that is completed, unexpired and unexposed. Then if the adversary is given either the session key (in this case let $b = 0$) or a random string (in this case let $b = 1$), each with probability* $1/2$,

*the probability of the adversary correctly guessing which one it received (i.e. correctly guessing the value of b) is not greater than 1/2 plus a negligible function in the security parameter.*

Two adversarial models are defined: the unauthenticated-links adversarial model (UM) and the authenticated-links adversarial model (AM). The only difference between the two is the amount of control the adversary has over the communications lines between principals. The UM corresponds to the "real world" where the adversary completely controls the network in use, and may modify or create messages from any party to any other party. The AM is a restricted version of the UM where the adversary may choose whether or not to deliver a message, but if a message is delivered, it must have been created by the specified sender and be delivered to the specified recipient without alteration. In addition, any such message may only be delivered once. In this way, authentication mechanisms can be separated from key agreement mechanisms by proving the key agreement secure in the AM, and then applying an authentication mechanism to the key agreement messages so that the overall protocol is secure in the UM.

An *authenticator* is a protocol translator that takes an SK-secure protocol in the AM to an SK-secure protocol in the UM. Authenticators can be constructed using one or more *message transmission (MT-) authenticators*. An MT-authenticator is a protocol which delivers one message in the UM in an authenticated manner. To translate an SK-secure protocol in the AM to an SK-secure protocol in the UM an MT-authenticator can be applied to each message and the resultant sub-protocols combined to form one overall SK-secure protocol in the UM. However, if the SK-secure protocol in the AM consists of more than one message, the resultant protocol is usually optimized to reduce the number and size of messages, involving reorder and reuse of message components. This practice was used in the CK-model proposal [9], although without a formal security proof.

The CK-model automatically ensures that secure protocols also provide perfect forward secrecy [9] through the use of session expiration. The CK-model also ensures that secure protocols are immune to unknown key share attacks [7, pp. 139–140]. This can be shown by contradiction. Suppose that an unknown key share attack exists on an SK-secure protocol. Let the attack proceed by convincing party $A$ that its key is shared with $D$, when in reality the key is shared with $B$. Then the two sessions in question can be identified as $(A, D, s)$ and $(B, A, s)$ where $s$ is the session identifier. These sessions are not matching (due to the different identities of the supposed participants), so it is possible for the adversary to choose one of the sessions as the test session and reveal the session key of the other session and hence break the security of the protocol, thus contradicting the original assumption. Therefore, SK-secure protocols also resist unknown key share attacks. Key compromise impersonation attacks [7, p. 52] are not covered by the CK-model since parties are unable to send or receive messages after corruption (it is only possible for the adversary to send or receive on the behalf of the corrupted party).

## 3 Definition of Secure Tripartite Key Exchange

The definitions of key exchange protocols and SK-security in the AM and UM provided by Canetti and Krawczyk in [9] are restricted to the case of two participants. It is necessary to extend the existing definitions to cater for at least three parties for use with tripartite key exchange. Therefore, the input to a key exchange protocol running within each party with identity $P_i$ is redefined to be $(D, sid, \texttt{role})$, where $sid$ is the session identifier, $D = \{P_i, P_j, P_k, \ldots\}$ is the set of identities of participants in the key exchange and $P_i \in D$. We make the new requirement that one and only one publicly available function, $f$, be specified (for the purpose of linking party identities to session identifiers). It is then required that for all inputs of the form $(D, sid, \texttt{role})$ to key exchange protocols, $f(D, s, d) = 1$ for some $d$, and $f(D', s, d') = 0$ for any set $D' \neq D$

and any $d'$ (including $d' = d$). It is also required that *sid* be unique to each party using it. As an example, let $D = \{P_i, P_j, P_k\}$, let $H$ be a collision resistant hash function, and let $N_i$, $N_j$ and $N_k$ be nonces freshly generated by $P_i$, $P_j$ and $P_k$ respectively, where the nonces are of a sufficient length that the probability of any one of them previously having been generated is negligible. Defining $s = H(P_i \parallel P_j \parallel P_k \parallel N_i \parallel N_j \parallel N_k)$ (where $\parallel$ indicates concatenation) and defining $f(\{P_i, P_j, P_k\}, s, (N_i \parallel N_j \parallel N_k)) = 1$ if and only if $s = H(P_i \parallel P_j \parallel P_k \parallel N_i \parallel N_j \parallel N_k)$ satisfies all of the above requirements.

**Definition 2 (Matching)** *Any u sessions (where each session is run by a different party) are* matching *if each session has the same session identifier. In particular, any two sessions with the same session identifier are said to be matching.*

**Definition 3 (Session key security)** *A t-party* KE *protocol $\pi$ is called* session key (SK-) secure *in the* AM *(respectively* UM*) if the following two properties hold for any adversary $\mathcal{A}$ (respectively $\mathcal{U}$) in the* AM *(respectively* UM*).*

1. *Protocol $\pi$ satisfies the property that if t uncorrupted parties complete a set of t matching sessions then they all output the same key.*
2. *The probability that $\mathcal{A}$ (respectively $\mathcal{U}$) guesses correctly the bit b from the test-session (i.e. outputs $b' = b$) is no more than $1/2$ plus a negligible function in the security parameter. (That is, it can do no better than randomly guess which one it received.)*

The two requirements of the definition of SK-security in the $t$ party case directly correspond to the two requirements of SK-security in the two party case. The modified requirements regarding the session identifier are the major change in the case of key exchange involving more than two parties. The identities of the participating parties are linked to the session identifier to make it easy to avoid scenarios where two or more sessions have identical keys but different session identities due to different beliefs by the sessions about who is participating in the protocol .

It is worth noting that in the CK-model, since uncorrupted protocol participants always follow the protocol, a protocol participant, say $A$, can trust another participant, say $B$, to pass on correct input from a third party, say $C$. The first party, $A$, does not need to receive an authenticated message from $C$, but only an authenticated message from $B$. $A$ trusts that $B$ received an authenticated message from $C$ containing the information that $B$ forwarded to $A$.

# 4 Tripartite Key Exchange Protocol in the AM

The notation used by the tripartite key exchange protocol is as follows:

| | |
|---|---|
| $A, B, C$ : | Protocol participants exchanging a secret key. |
| $P$ : | Base point of the elliptic curve. |
| $\mathbb{G}_1$ : | Points on an elliptic curve with a suitable pairing. |
| $\mathbb{G}_2$ : | Group of the same size as $\mathbb{G}_1$. |
| $n$ : | Order of $\mathbb{G}_1$ and $\mathbb{G}_2$. |
| $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ : | An admissible bilinear map (the properties of such a map are below). |
| $\sigma_X^Y$ : | Signature by $X$ intended for $Y$. Specifying the intended recipient clarifies the purpose of each signature in protocol descriptions, although in practice the intended recipient need not be specified. |
| ${}^X\mathcal{E}_Y$ : | Encryption by $X$ intended for $Y$. Specifying the sender clarifies the purpose of each encryption in protocol descriptions, although in practice the sender does not necessarily need to be specified. |

An admissible bilinear map must be bilinear, non-degenerate and computable [5]. That is, the map must satisfy $e([a]P, [b]Q) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$, the map must not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$, and there must be an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in \mathbb{G}_1$. It is possible to construct an admissible bilinear map based on either the Weil pairing or Tate pairing over an elliptic curve [12, 5].

Joux has described an unauthenticated broadcast tripartite key exchange protocol which requires only one round [13, 19], shown in Protocol 1. This protocol can be used as a building block in the CK-model to form authenticated tripartite key exchange protocols. However, because there are no authenticators available for broadcast protocols, proving this version of the protocol secure in the AM is not a useful exercise. Therefore, Protocol 1 has been modified here to create the unicast version of the protocol in the AM shown by Protocol 2. The messages in this version are almost identical to those of Protocol 1, the only difference being the addition of the session identifier, *sid*, in all messages. The value of *sid* is not specified here, but the CK-model assumes it to be known by protocol participants before the protocol begins. In practice, the session identifier may be determined during protocol execution [9, 20]. It is assumed that messages in the AM implicitly specify sender and receiver. If it is not possible to determine the identities of all protocol participants from *sid* (e.g. the case where *sid* contains a hash of the identities), it may be necessary to include the identities of the participating parties in the first two messages from $A$. However, since all parties must ensure the correctness of *sid*, such "hints" can be omitted from the formal protocol specification.

$$
\begin{array}{lll}
A \rightarrow B, C: & [a]P & a \in_R \mathbb{Z}_n \\
B \rightarrow A, C: & [b]P & b \in_R \mathbb{Z}_n \\
C \rightarrow A, B: & [c]P & c \in_R \mathbb{Z}_n \\
\multicolumn{3}{l}{Key: e(P,P)^{abc} = e([b]P,[c]P)^a = e([a]P,[c]P)^b = e([a]P,[b]P)^c}
\end{array}
$$

**Protocol 1:** Joux broadcast protocol

| | |
|---|---|
| $A$ on input $(A, B, C, sid)$: | $C$ on receipt of $(sid, [a]P)$: |
| $A \rightarrow B$: $(sid, [a]P)$, $a \in_R \mathbb{Z}_n$ | $C \rightarrow A$: $(sid, [c]P)$, $c \in_R \mathbb{Z}_n$ |
| $A \rightarrow C$: $(sid, [a]P)$ | $C \rightarrow B$: $(sid, [c]P)$ |
| $B$ on receipt of $(sid, [a]P)$: | Shared Key: $e(P,P)^{abc} = e([b]P,[c]P)^a$ |
| $B \rightarrow A$: $(sid, [b]P)$, $b \in_R \mathbb{Z}_n$ | $= e([a]P,[c]P)^b$ |
| $B \rightarrow C$: $(sid, [b]P)$ | $= e([a]P,[b]P)^c$ |

**Protocol 2:** Joux protocol in the AM without broadcast messages

The six messages of Protocol 2 can be reduced to four messages by allowing one party to act as messenger between the other two parties. Such a protocol has been created in the course of this research and is shown by Protocol 3.

In order to prove the security of Protocols 2 and 3 in the AM, it is necessary to assume that the Decisional Bilinear Diffie-Hellman Problem (DBDH) is hard. The assumption has been studied in [11], and can be described similarly to the Decisional Diffie-Hellman assumption of [9] as follows:

**Definition 4 (DBDH assumption)** *Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear map that takes as input two elements of $\mathbb{G}_1$ and outputs an element of $\mathbb{G}_2$. Let $n$ be the order of $\mathbb{G}_1$ and $\mathbb{G}_2$, and let $P$ be an element of $\mathbb{G}_1$. Let two probability distributions of tuples of seven elements, $Q_0$ and $Q_1$, be defined as:*

$B \rightarrow A: \quad (sid, [b]P) \quad (\text{where } b \in_R \mathbb{Z}_n)$
$C \rightarrow A: \quad (sid, [c]P) \quad (\text{where } c \in_R \mathbb{Z}_n)$
$A \rightarrow B: \quad (sid, [a]P, [c]P) \quad (\text{where } a \in_R \mathbb{Z}_n)$
$A \rightarrow C: \quad (sid, [a]P, [b]P)$
$Key: e(P,P)^{abc} = e([b]P, [c]P)^a = e([a]P, [c]P)^b = e([a]P, [b]P)^c$

**Protocol 3:** Variant of Joux protocol in the AM that can be used with authenticators to create efficient UM protocols

$$Q_0 = \{\langle \mathbb{G}_1, \mathbb{G}_2, P, [a]P, [b]P, [c]P, e(P,P)^{abc} \rangle : a, b, c \in_R \mathbb{Z}_n\} \text{ and}$$
$$Q_1 = \{\langle \mathbb{G}_1, \mathbb{G}_2, P, [a]P, [b]P, [c]P, e(P,P)^{d} \rangle : a, b, c, d \in_R \mathbb{Z}_n\}.$$

*Then the* DBDH *assumption states that* $Q_0$ *and* $Q_1$ *are computationally indistinguishable.*

**Theorem 1** *Given the* DBDH *assumption, Protocols 2 and 3 are both* SK*-secure in the* AM*.*

The proof of Theorem 1 is provided in Appendix B. It is possible to modify the protocol so that the use of the DBDH assumption in the proof can be replaced with the use of a random oracle. This also requires the proof to use the assumption that the Bilinear Diffie-Hellman (BDH) problem is hard. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear map that takes as input two elements of $\mathbb{G}_1$ and outputs an element of $\mathbb{G}_2$. Let $n$ be the order of $\mathbb{G}_1$ and $\mathbb{G}_2$, and let $P$ be an element of $\mathbb{G}_1$. Then the BDH problem [5] is to find $e(P,P)^{abc}$ when given $(\mathbb{G}_1, \mathbb{G}_2, P, [a]P, [b]P, [c]P)$, where $a$, $b$ and $c \in_R \mathbb{Z}_n$. If the BDH problem is hard, there is no polynomial time algorithm to solve the BDH problem with non-negligible probability.

One way to modify the protocol to use this proof method is to combine $e(P,P)^{abc}$ with some sort of hash function to produce the key (e.g. $H\left(e\left(P,P\right)^{abc}\right)$ or a keyed hash function $H_{e(P,P)^{abc}}([a]P, [b]P, [c]P)$). The logic of the proof is based on the observation that since the hash function is completely random, the adversary can only obtain information about the session key by querying the hash function oracle with the input that would have been used to generate the session key. However, if the adversary is able to produce such a value with which to query the oracle, then the adversary is also able to break the BDH problem, which was assumed to be hard. The formal proof proceeds in a similar fashion to that of the proof using the DBDH assumption.

## 5 Applying Authenticators to the Joux Protocol

In order to create an SK-secure protocol in the UM, it is necessary to apply one or more authenticators to the Joux protocol. Here we focus on two authenticators originally proposed by Bellare et al. [3], $\lambda_{\text{SIG}}$ (requiring the use of a signature scheme secure against adaptive chosen message attacks [17]) and $\lambda_{\text{ENC}}$ (requiring the use of an encryption scheme indistinguishable under chosen ciphertext attacks [4] and a secure MAC scheme). Their specifications are given by Protocols 10 and 11 in Appendix A .

Applying $\lambda_{\text{SIG}}$ to each message of the Joux protocol (Protocol 2) results in Protocol 12 in Appendix A. However, it is possible to optimize this protocol to produce a much more efficient version. This can be done by using $[a]P$ in place of $r_A$ and $r'_A$, $[b]P$ in place of $r_B$ and $r'_B$, and $[c]P$ in place of $r_C$ and $r'_C$ to avoid creating and transmitting these extra nonces. In addition, in most cases, the two signatures produced by each party can be combined to a single signature containing one copy of each of the items originally contained in the two separate signatures. Finally, only the session identifier needs to be included at the beginning of each UM message to

determine to which session the messages belong. (In the specification of the MT-authenticators, the messages were unique and the entire message from the AM was included at the start of each UM message for this purpose since there were no session identifiers.) The resultant protocol in the UM is shown by Protocol 4 and requires a total of five messages and four signatures.

$$
\begin{array}{ll}
A \rightarrow B: & (sid, [a]P) \quad (\text{where } a \in_R \mathbb{Z}_n) \\
B \rightarrow C: & \left(sid, [a]P, [b]P, \sigma_B^A\left(A, sid, [a]P, [b]P\right)\right) \quad (\text{where } b \in_R \mathbb{Z}_n) \\
C \rightarrow A: & \left(sid, [b]P, [c]P, \sigma_C^{A,B}\left(A, B, sid, [a]P, [b]P, [c]P\right), \sigma_B^A\left(A, sid, [a]P, [b]P\right)\right) \\
 & \qquad\qquad (\text{where } c \in_R \mathbb{Z}_n) \\
A \rightarrow B: & \left(\ sid, [c]P, \sigma_A^{B,C}\left(B, C, sid, [a]P, [b]P, [c]P\right), \sigma_C^{A,B}\left(A, B, sid, [a]P, [b]P, [c]P\right)\ \right) \\
B \rightarrow C: & \left(\sigma_B^C\left(C, sid, [b]P, [c]P\right), \sigma_A^{B,C}\left(B, C, sid, [a]P, [b]P, [c]P\right)\right) \\
 & Key: e(P,P)^{abc} = e([b]P, [c]P)^a = e([a]P, [c]P)^b = e([a]P, [b]P)^c
\end{array}
$$

**Protocol 4:** Joux protocol authenticated with $\lambda_{\text{SIG}}$

It is possible to combine $\sigma_B^A(A, sid, [a]P, [b]P)$ and $\sigma_B^C(C, sid, [b]P, [c]P)$ from Protocol 4 into one signature at the expense of an extra message, as shown by Protocol 5. Protocol 6 is another possible UM protocol where some messages have been combined after the authenticator has been applied to create a broadcast protocol. It has five broadcasts and three signatures.

$$
\begin{array}{ll}
A \rightarrow B: & (sid, [a]P) \quad (\text{where } a \in_R \mathbb{Z}_n) \\
B \rightarrow C: & (sid, [a]P, [b]P) \quad (\text{where } b \in_R \mathbb{Z}_n) \\
C \rightarrow A: & \left(sid, [b]P, [c]P, \sigma_C^{A,B}\left(A, B, sid, [a]P, [b]P, [c]P\right)\right) (\text{where } c \in_R \mathbb{Z}_n) \\
A \rightarrow B: & \left(\ sid, [c]P, \sigma_A^{B,C}\left(B, C, sid, [a]P, [b]P, [c]P\right), \sigma_C^{A,B}\left(A, B, sid, [a]P, [b]P, [c]P\right)\ \right) \\
B \rightarrow C: & \left(\sigma_B^{A,C}\left(A, C, sid, [a]P, [b]P, [c]P\right), \sigma_A^{B,C}\left(B, C, sid, [a]P, [b]P, [c]P\right)\right) \\
B \text{ or } C \rightarrow A: & \sigma_B^{A,C}\left(A, C, sid, [a]P, [b]P, [c]P\right) \\
 & Key: e(P,P)^{abc} = e([b]P, [c]P)^a = e([a]P, [c]P)^b = e([a]P, [b]P)^c
\end{array}
$$

**Protocol 5:** Joux protocol authenticated with $\lambda_{\text{SIG}}$ using a minimal number of signatures

$$
\begin{array}{ll}
A \rightarrow B, C: & (sid, [a]P) \quad (\text{where } a \in_R \mathbb{Z}_n) \\
B \rightarrow C, A: & (sid, [b]P) \quad (\text{where } b \in_R \mathbb{Z}_n) \\
C \rightarrow A, B: & \left(sid, [c]P, \sigma_C^{A,B}\left(sid, A, B, [a]P, [b]P, [c]P\right)\right) \quad (\text{where } c \in_R \mathbb{Z}_n) \\
A \rightarrow B, C: & \left(sid, \sigma_A^{B,C}\left(B, C, sid, [a]P, [b]P, [c]P\right)\right) \\
B \rightarrow A, C: & \left(\sigma_B^{C,A}\left(A, C, sid, [a]P, [b]P, [c]P\right)\right) \\
 & Key: e(P,P)^{abc} = e([b]P, [c]P)^a = e([a]P, [c]P)^b = e([a]P, [b]P)^c
\end{array}
$$

**Protocol 6:** Joux protocol authenticated with $\lambda_{\text{SIG}}$, broadcast version

The $\lambda_{\text{SIG}}$ authenticator can be applied to Protocol 3 to produce Protocol 13 in Appendix A. This protocol can be optimized in a similar way to Protocol 12 to produce a protocol in the UM which requires five messages but only three signatures, shown as Protocol 7.

A protocol resulting from applying the $\lambda_{\text{ENC}}$ authenticator to the AM Joux protocol (Protocol 2) is described by Protocol 14 in Appendix A and an optimized version is described by Protocol 8. The optimized protocol requires a total of five messages, six encryptions and six MACs. Allowing messages to be broadcast does not change these requirements.

$$
\begin{array}{rl}
A \to B: & (sid, [a]P) \quad (\text{where } a \in_R \mathbb{Z}_n) \\
B \to C: & \left(sid, [a]P, [b]P, \sigma_B^A\left(A, sid, [a]P, [b]P\right)\right) \quad (\text{where } b \in_R \mathbb{Z}_n) \\
C \to A: & \left(sid, [b]P, [c]P, \sigma_C^A\left(A, sid, [a]P, [c]P\right), \sigma_B^A\left(A, sid, [a]P, [b]P\right)\right) \quad (\text{where } c \in_R \mathbb{Z}_n) \\
A \to B: & \left(sid, [c]P, \sigma_A^{B,C}\left(B, C, sid, [a]P, [b]P, [c]P\right)\right) \\
A \text{ or } B \to C: & \left(sid, [c]P, \sigma_A^{B,C}\left(B, C, sid, [a]P, [b]P, [c]P\right)\right) \\
& Key : e(P,P)^{abc} = e([b]P,[c]P)^a = e([a]P,[c]P)^b = e([a]P,[b]P)^c
\end{array}
$$

**Protocol 7:** Variant of Joux protocol authenticated with $\lambda_{\text{SIG}}$

$$
\begin{array}{rl}
A \to B: & sid, [a]P, \ {}^A\mathcal{E}_B(N_{AB}), \ {}^A\mathcal{E}_C(N_{AC}) \\
B \to C: & sid, [a]P, [b]P, \ {}^B\mathcal{E}_C(N_{BC}), \ {}^B\mathcal{E}_A(N_{BA}), \text{MAC}_{N_{AB}}(sid, [b]P, A), \ {}^A\mathcal{E}_C(N_{AC}) \\
C \to A: & sid, [b]P, [c]P, \ {}^C\mathcal{E}_A(N_{CA}), \ {}^C\mathcal{E}_B(N_{CB}), \text{MAC}_{N_{AC}}(sid, [c]P, A), \\
& \text{MAC}_{N_{BC}}(sid, [c]P, B), \ {}^B\mathcal{E}_A(N_{BA}), \text{MAC}_{N_{AB}}(sid, [b]P, A) \\
A \to B: & sid, [c]P, \text{MAC}_{N_{BA}}(sid, [a]P, B), \text{MAC}_{N_{CA}}(sid, [a]P, C), \ {}^C\mathcal{E}_B(N_{CB}), \\
& \text{MAC}_{N_{BC}}(sid, [c]P, B) \\
B \to C: & sid, \text{MAC}_{N_{CB}}(sid, [b]P, C), \text{MAC}_{N_{CA}}(sid, [a]P, C) \\
& Key : e(P,P)^{abc} = e([b]P,[c]P)^a = e([a]P,[c]P)^b = e([a]P,[b]P)^c
\end{array}
$$

**Protocol 8:** Joux protocol authenticated with $\lambda_{\text{ENC}}$

Another protocol using $\lambda_{\text{ENC}}$ can be constructed in the UM, by using the variant of the Joux protocol in the AM (Protocol 3). The unoptimized protocol is shown by Protocol 15 in Appendix A. The optimized version is shown by Protocol 9 and requires five messages, four encryptions and four MACs. In a broadcast version of the protocol, the last two messages can be combined into one broadcast so that only four messages are required. However, the same number of encryptions and MACs are still required by the broadcast version.

$$
\begin{array}{rl}
A \to B: & sid, [a]P, \ {}^A\mathcal{E}_B(N_{AB}), \ {}^A\mathcal{E}_C(N_{AC}) \\
B \to C: & sid, [a]P, [b]P, \ {}^B\mathcal{E}_A(N_{BA}), \ \text{MAC}_{N_{AB}}(sid, [b]P, A), \ {}^A\mathcal{E}_C(N_{AC}) \\
C \to A: & sid, [b]P, [c]P, \ {}^C\mathcal{E}_A(N_{CA}), \ \text{MAC}_{N_{AC}}(sid, [c]P, A), \\
& \quad {}^B\mathcal{E}_A(N_{BA}), \ \text{MAC}_{N_{AB}}(sid, [b]P, A) \\
A \to B: & sid, [c]P, \ \text{MAC}_{N_{BA}}(sid, C, [a]P, [c]P, B), \ \text{MAC}_{N_{CA}}(sid, [a]P, [b]P, C) \\
A/B \to C: & sid, \ \text{MAC}_{N_{CA}}(sid, [a]P, [b]P, C) \\
& Key : e(P,P)^{abc} = e([b]P,[c]P)^a = e([a]P,[c]P)^b = e([a]P,[b]P)^c
\end{array}
$$

**Protocol 9:** Variant of Joux protocol authenticated with $\lambda_{\text{ENC}}$

# 6 Efficiency of Joux Based Protocols in the UM

Table 1 shows the efficiency of each of the different optimized protocols in the UM described in Section 5. The table shows that the efficiency of each scheme depends heavily on the signature or encryption scheme chosen for the implementation. Since the protocols will be executed using an elliptic curve where a pairing is available that can be used as the basis of an admissible bilinear map (as defined at the beginning of Section 4), the suitability of various pairing-based signature and encryption schemes for the above protocols has been investigated. A brief description of each scheme is included below, and the efficiency of each scheme summarized in Tables 2 and 3.

**M-L** This identity-based signcryption scheme was proposed by Malone-Lee [16]. A summary of its efficiency is also provided by Nalla and Reddy [18]. The scheme provides non-

Table 1: Operations and messages required by tripartite UM protocols

| Protocol number | 4 | 5 | 7 | 7 | 6 | 8 | 9 | 9 |
|---|---|---|---|---|---|---|---|---|
| Broadcast used | N | N | N | Y | Y | N/Y | N | Y |
| Messages | 5 | 6 | 5 | 4 | 5 | 5 | 5 | 4 |
| Signatures | 4 | 3 | 3 | 3 | 3 | - | - | - |
| Verifications | 6 | 6 | 4 | 4 | 6 | - | - | - |
| Encryptions | - | - | - | - | - | 6 | 4 | 4 |
| Decryptions | - | - | - | - | - | 6 | 4 | 4 |
| MACs | - | - | - | - | - | 6 | 4 | 4 |
| Scalar mults. | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| Exponentiations | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| Pairings | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |

Table 2: Efficiency of signature schemes using pairings

| Scheme | Signature | | | | Verification | | | |
|---|---|---|---|---|---|---|---|---|
| | Pair. | Exp. | Sc. mul. | Other | Pair. | Exp. | Sc. mul. | Other |
| Hess | - | 1 | 1 | | 1+(1) | 1 | - | |
| BLS | - | - | 1 | | 2 | - | - | |
| CC | - | - | 2 | | 2 | - | 1 | |
| LQ | (1) | 2 | 1.4 | symm. enc. | 2+(2) | 1 | - | symm. dec. |
| M-L | (1) | - | 3 | | 3+(1) | 1 | - | |
| SOK | - | - | 2 | | 2 or 3 | - | - | |

$(y)$ indicates an additional $y$ operations required in a precomputation.

repudiation if the plaintext is surrendered to the party required to perform an independent verification. The scheme can therefore be used in place of a signature scheme if desired.

**NR** This identity-based encryption scheme (requiring a trusted authority) was proposed by Nalla and Reddy [18]. The trusted authority can also be used to provide non-repudiation.

**BLS** This scheme to provide short signatures was proposed by Boneh, Lynn and Shacham [6]. The scheme is not identity-based and allows different signatures to be combined, thus saving bandwidth (at the expense of extra computation). The scheme can alse be used for batch verification, to increase verification efficiency if several users sign the same message.

**Lynn** This scheme to provide authenticated identity-based encryption was proposed by Lynn [15]. The scheme does not provide non-repudiation [14] and so can not be used in place of a signature scheme. It is noteworthy that this scheme actually uses fewer pairings than the BF scheme which provides encryption only. However, the Lynn scheme does require use of symmetric encryption and decryption algorithms.

**CC** This scheme to provide an identity-based signature was proposed by Cha and Cheon [10].

**Hess** This scheme was proposed by Hess [12] and is an identity-based signature scheme. The paper also includes a comparison with the CC and SOK schemes.

**SOK** This scheme was proposed by Sakai, Ohgishi and Kasahara and an efficiency analysis is provided by Hess [12].

**BF** This identity-based encryption scheme was proposed by Boneh and Franklin [5].

**LQ** This identity-based signcryption scheme was proposed by Libert and Quisquater [14]. It can require the use of a symmetric encryption and decryption scheme, and can be used as either a signature or an encryption scheme since it provides non-repudiation because any party can verify the origin of the ciphertext. However, verification of the origin of the plaintext requires the key used for the symmetric encryption to be provided to the party performing the verification. Another property of the scheme is that the symmetric encryption and decryption can be replaced by some extra modular multiplications if the plaintext to be encrypted is only short. The signcryption requires a total of two scalar multiplications, but these can be performed together in the time of about 1.4 scalar multiplications.

Although the signcryption schemes can be used as either a signature or encryption schemes, care must be taken when performing an efficiency analysis of the resulting UM protocol, since extra signatures may need to be created if a single signature was intended for use by more than one recipient in the original UM protocol.

Since the pairing operation is the most expensive of those performed by the signature and encryption schemes under consideration, the authenticated identity-based encryption scheme of Lynn appears to be the most promising from an efficiency viewpoint. Combining it with the protocol requiring the least number of operations, Protocol 9, leads to an implementation of the Joux protocol in the UM requiring three on-line pairings to compute the key (one per party) and four off-line pairings. Four instead of eight off-line pairings are required since some of the off-line pairings can be reused and need not be calculated twice.

Table 4 provides a comparison of the number of operations required by Protocol 9 and those required by the tripartite protocols proposed by Al-Riyami and Paterson [1] and based on Joux's protocol, TAK-1 to TAK-4 and TAKC. The table shows that those protocols using broadcast

Table 3: Efficiency of encryption schemes using pairings

| Scheme | Encryption | | | | Decryption | | | |
|---|---|---|---|---|---|---|---|---|
| | Pair. | Exp. | Sc. mul. | Other | Pair. | Exp. | Sc. mul. | Other |
| Lynn | (1) | - | - | symm. enc. | (1) | - | - | symm. dec. |
| BF | (1) | 1 | 1 | | 1 | - | 1 | |
| NR | (1) | 1 | 2 | | 2+(1) | 1 | - | |
| | | | | option | | | | option |
| LQ | (1) | 2 | 1.4 | symm. enc. | 2+(2) | 1 | - | symm. dec. |
| M-L | (1) | - | 3 | | 3+(1) | 1 | - | |

$(y)$ indicates an additional $y$ operations required in a precomputation.

Table 4: Operations and messages required by Al-Riyami and Paterson's tripartite protocols compared with Protocol 9

| Protocol name | TAK-1 | TAK-2 | TAK-3 | TAK-4 | TAKC | 9 | 9 |
|---|---|---|---|---|---|---|---|
| Broadcast used | Y | Y | Y | Y | N | N | Y |
| Messages | 3 | 3 | 3 | 3 | 6 | 5 | 4 |
| Signatures | - | - | - | - | 3 | - | - |
| Verifications | - | - | - | - | 6 | - | - |
| Symmetric encryptions | - | - | - | - | 3 | 4 | 4 |
| Symmetric decryptions | - | - | - | - | 6 | 4 | 4 |
| MACs | - | - | - | - | - | 4 | 4 |
| Scalar mults. | [3] | [3] | [3] | 6 + [3] | [3] | [3] | [3] |
| Exponentiations | $3 + \langle 3 \rangle$ | 6 | $3 + \langle 3 \rangle$ | 3 | 3 | 3 | 3 |
| Pairings | $3 + \langle 3 \rangle$ | 9 | $6 + \langle 3 \rangle$ | 3 | 3 | $3 + (4)$ | $3 + (4)$ |

$y + \langle x \rangle$ indicates a total of $y + x$ operations are required, but $x$ operations may be precomputed if identities and long term keys of participants known in advance. A new precomputation is required for each key exchange.

$y + (x)$ indicates a total of $y + x$ operations are required, but $x$ operations may be precomputed if identities of participants known in advance. The precomputation is reusable for any key exchange involving those participants.

$[x]$ indicates that $x$ operations may be precomputed, but a new precomputation is required for each key exchange.

messages (TAK-1 to TAK-4) only require 3 messages, which is less than the most efficient of the protocols proposed here. However, such protocols do not provide message authentication, only implicit key authentication. Protocol 9 has the advantage that parties accepting a secret key can be sure that the messages upon which they acted were not generated by a malicious party or replays of old messages; the other parties actually participated in the key exchange.

It is also possible to compare Protocol 9 with existing schemes for group key exchange based on the ordinary use of discrete logarithms, such as that of Bresson, Chevassut, Pointcheval and Quisquater [8], herein denoted the BCPQ scheme. This scheme can be converted to a tripartite key exchange protocol requiring eight exponentiations (two of which can be precomputed), three signatures and four verifications. If a signature scheme such as DSA is used, signing takes one exponentiation (which can be precomputed) and verification takes two simultaneous exponentiations, or about the time of 1.2 single exponentiations. Thus the BCPQ scheme takes the total time of 10.8 online exponentiations and 5 offline exponentiations, whereas Protocol 9 requires 3 exponentiations and 3 pairings online. Therefore, if a pairing can be computed in the time of 2.6 exponentiations, the Joux based scheme will be as efficient in terms of online

computation as the BCPQ scheme. Figures due to Barreto, Kim, Lynn and Scott [2] indicate that a 512 bit pairing takes about 2.5 times as long as a 1024 bit exponentiation with a 1007 bit exponent (20ms for a pairing compared to 7.9ms for an RSA signature) or 4.9 times as long as a 1024 bit exponentiation with a 160 bit exponent (20ms for a pairing compared to 4.09ms for a DSA signature). Thus Protocol 9 compares favourably to the BCPQ scheme if a large exponent is used with that scheme, but not if a small exponent is used. However, there has recently been a substantial amount of research on improving pairing efficiency, and it is possible that the efficiency of pairings may improve to the extent that Protocol 9 is more efficient than the BCPQ scheme for small exponents also.

# 7 Conclusion

The CK-model has been used to examine the security of tripartite key exchange protocols based on the Joux protocol. A new definition of security for key exchange protocols with more than two participants has been provided, and a proof of security for the Joux protocol in the AM given. The efficiency of the UM protocols created by combining the Joux AM protocol with signature and encryption based authenticators has been analysed, and the efficiency of various pairing based encryption and signature schemes which could be used in the authentication mechanism has been summarized. It has been concluded that a secure tripartite key exchange protocol can be formed that requires three on-line and four off-line pairings. This protocol also compared favourably with other published tripartite key agreement protocols.

# References


[1] Sattam S. Al-Riyami and Kenneth G. Paterson. Tripartite authenticated key agreement protocols from pairings. In *Cryptography and Coding*, volume 2898 of *Lecture Notes in Computer Science*, pages 332–359. Springer-Verlag, 2003.

[2] Paulo S. L. M. Barreto, Hae Y. Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology—CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer-Verlag, 2002.

[3] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC '98)*, pages 419–428, New York, May 1998. ACM Press. [Full paper online] `http://www-cse.ucsd.edu/users/mihir/papers/modular.ps.gz`.

[4] Mihir Bellare, Anand Desai, David Pointcheval, and Phil Rogaway. Relations among notions of security for public-key encryption schemes (extended abstract). In *Advances in Cryptology—CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer-Verlag, 1998. [Full paper online] `http://www-cse.ucsd.edu/users/mihir/papers/relations.pdf`.

[5] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology—CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, 2001. [Full paper online] `http://crypto.stanford.edu/~dabo/abstracts/ibe.html`.

[6] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In *Advances in Cryptology—ASIACRYPT 2001*, volume 2139 of *Lecture Notes*

*in Computer Science*, pages 514–532. Springer-Verlag, 2001. [Full paper online] `http://crypto.stanford.edu/~dabo/abstracts/weilsigs.html`.

[7] Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*. Springer-Verlag, Berlin, 2003.

[8] E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater. Provably authenticated group Diffie-Hellman key exchange. In P. Samarati, editor, *Proc. of ACM-CCS 01*, pages 255–264, Philadelphia, Pennsylvania, USA, November 2001. ACM, ACM Press.

[9] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *Advances in Cryptology—EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 451–472. Springer-Verlag, 2001. [Full paper online] `http://eprint.iacr.org/2001/040.ps.gz`.

[10] Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap Diffie-Hellman groups. In *Practice and Theory in Public Key Cryptography—PKC 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30. Springer-Verlag, 2003.

[11] Jung Hee Cheon and Dong Hoon Lee. Diffie-Hellman problems and bilinear maps. Cryptology ePrint Archive, Report 2002/117, 2002. [Online] `http://eprint.iacr.org/` [accessed 11/07/2003].

[12] Florian Hess. Efficient identity based signature schemes based on pairings. In *Selected Areas in Cryptography—SAC 2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 310–324. Springer-Verlag, 2002.

[13] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. In *Algorithmic Number Theory: Fourth International Symposium—ANTS-IV 2000, Proceedings*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–393. Springer-Verlag, 2000.

[14] Benoît Libert and Jean-Jacques Quisquater. New identity based signcryption schemes from pairings. Cryptology ePrint Archive, Report 2003/023, 2003. [Online] `http://eprint.iacr.org/` [accessed 11/07/2003].

[15] Ben Lynn. Authenticated identity-based encryption. Cryptology ePrint Archive, Report 2002/072, 2002. [Online] `http://eprint.iacr.org/` [accessed 11/07/2003].

[16] John Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. [Online] `http://eprint.iacr.org/` [accessed 11/07/2003].

[17] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[18] Divya Nalla and K.C. Reddy. Signcryption scheme for identity-based cryptosystems. Cryptology ePrint Archive, Report 2003/066, 2003. [Online] `http://eprint.iacr.org/` [accessed 11/07/2003].

[19] K.G. Paterson. Cryptography from pairings: a snapshot of current research. *Information Security Technical Report*, 7(3):41–54, 2002.

[20] Yiu Shing Terry Tin, Colin Boyd, and Juan Manuel González Nieto. Provably secure mobile key exchange: Applying the Canetti-Krawczyk approach. In *Information Security and Privacy—ACISP 2003*, volume 2727 of *Lecture Notes in Computer Science*, pages 166–179. Springer-Verlag, 2003.

# A Authenticators and Unoptimized Protocols

This appendix contains two previously published authenticators, as well as unoptimized versions of the protocols described in this paper. The protocols are generated by applying one of the authenticators to each message of a protocol that has been proven secure in the AM.

**Protocol 10:** $\lambda_{\text{ENC}}$

| $A$ | | $B$ |
|---|---|---|
| | $\underrightarrow{m}$ | |
| | | $N_B \in_R \{0,1\}^k$ |
| | $\underleftarrow{m,\ {}^B\mathcal{E}_A(N_B)}$ | |
| | $\underrightarrow{m, \text{MAC}_{N_B}(m,B)}$ | |

**Protocol 11:** $\lambda_{\text{SIG}}$

| $A$ | | $B$ |
|---|---|---|
| | $\underrightarrow{m}$ | |
| | | $r_B \in_R \{0,1\}^k$ |
| | $\underleftarrow{m, r_B}$ | |
| | $\underrightarrow{m, \sigma_A^B(m, r_B, B)}$ | |

**Protocols 10 and 11:** Encryption and signature-based MT-authenticators, $\lambda_{\text{ENC}}$ and $\lambda_{\text{SIG}}$

$$
\begin{array}{lll}
A \to B: & (sid, [a]P) & (\text{where } a \in_R \mathbb{Z}_n) \\
B \to A: & (sid, [a]P, r_B) & (\text{where } r_B \in_R \{0,1\}^k) \\
A \to B: & \left(sid, [a]P, \sigma_A^B\left(sid, [a]P, r_B, B\right)\right) & \\
A \to C: & (sid, [a]P) & \\
C \to A: & (sid, [a]P, r_C) & (\text{where } r_C \in_R \{0,1\}^k) \\
A \to C: & \left(sid, [a]P, \sigma_A^C\left(sid, [a]P, r_C, C\right)\right) & \\
B \to A: & (sid, [b]P) & (\text{where } b \in_R \mathbb{Z}_n) \\
A \to B: & (sid, [b]P, r_A) & (\text{where } r_A \in_R \{0,1\}^k) \\
B \to A: & \left(sid, [b]P, \sigma_B^A\left(sid, [b]P, r_A, A\right)\right) & \\
B \to C: & (sid, [b]P) & \\
C \to B: & (sid, [b]P, r'_C) & (\text{where } r'_C \in_R \{0,1\}^k) \\
B \to C: & \left(sid, [b]P, \sigma_B^C\left(sid, [b]P, r'_C, C\right)\right) & \\
C \to A: & (sid, [c]P) & (\text{where } c \in_R \mathbb{Z}_n) \\
A \to C: & (sid, [c]P, r'_A) & (\text{where } r'_A \in_R \{0,1\}^k) \\
C \to A: & \left(sid, [c]P, \sigma_C^A\left(sid, [c]P, r'_A, A\right)\right) & \\
C \to B: & (sid, [c]P) & \\
B \to C: & (sid, [c]P, r'_B) & (\text{where } r'_B \in_R \{0,1\}^k) \\
C \to B: & \left(sid, [c]P, \sigma_C^B\left(sid, [c]P, r'_B, B\right)\right) &
\end{array}
$$

$$Key : e(P,P)^{abc} = e([b]P,[c]P)^a = e([a]P,[c]P)^b = e([a]P,[b]P)^c$$

**Protocol 12:** Joux protocol authenticated with $\lambda_{\text{SIG}}$, not optimized

$$
\begin{array}{lll}
B \to A: & (sid, [b]P) & (\text{where } b \in_R \mathbb{Z}_n) \\
A \to B: & (sid, [b]P, r_A) & (\text{where } r_A \in_R \{0,1\}^k) \\
B \to A: & \left(sid, [b]P, \sigma_B^A\left(sid, [b]P, r_A, A\right)\right) & \\
C \to A: & (sid, [c]P) & (\text{where } c \in_R \mathbb{Z}_n) \\
A \to C: & (sid, [c]P, r'_A) & (\text{where } r'_A \in_R \{0,1\}^k) \\
C \to A: & \left(sid, [c]P, \sigma_C^A\left(sid, [c]P, r'_A, A\right)\right) & \\
A \to B: & (sid, [a]P, [c]P) & (\text{where } a \in_R \mathbb{Z}_n) \\
B \to A: & (sid, [a]P, [c]P, r_B) & (\text{where } r_B \in_R \{0,1\}^k) \\
A \to B: & \left(sid, [a]P, [c]P, \sigma_A^B\left(sid, [a]P, [c]P, r_B, B\right)\right) & \\
A \to C: & (sid, [a]P, [b]P) & \\
C \to A: & (sid, [a]P, [b]P, r_C) & (\text{where } r_C \in_R \{0,1\}^k) \\
A \to C: & \left(sid, [a]P, [b]P, \sigma_A^C\left(sid, [a]P, [b]P, r_C, C\right)\right) &
\end{array}
$$

$$Key : e(P,P)^{abc} = e([b]P,[c]P)^a = e([a]P,[c]P)^b = e([a]P,[b]P)^c$$

**Protocol 13:** Variant of Joux protocol authenticated with $\lambda_{\text{SIG}}$, not optimized

$$
\begin{array}{lll}
A \to B: & (sid, [a]P) & (\text{where } a \in_R \mathbb{Z}_n) \\
B \to A: & \left(sid, [a]P,\ {}^B\mathcal{E}_A(N_{BA})\right) & (\text{where } N_{BA} \in_R \{0,1\}^k) \\
A \to B: & (sid, [a]P, \text{MAC}_{N_{BA}}(sid, [a]P, B)) & \\
A \to C: & (sid, [a]P) & \\
C \to A: & \left(sid, [a]P,\ {}^C\mathcal{E}_A(N_{CA}\right) & (\text{where } N_{CA} \in_R \{0,1\}^k) \\
A \to C: & (sid, [a]P, \text{MAC}_{N_{CA}}(sid, [a]P, C)) & \\
B \to A: & (sid, [b]P) & (\text{where } b \in_R \mathbb{Z}_n) \\
A \to B: & \left(sid, [b]P,\ {}^A\mathcal{E}_B(N_{AB}\right) & (\text{where } N_{AB} \in_R \{0,1\}^k) \\
B \to A: & (sid, [b]P, \text{MAC}_{N_{AB}}(sid, [b]P, A)) & \\
B \to C: & (sid, [b]P) & \\
C \to B: & \left(sid, [b]P,\ {}^C\mathcal{E}_B(N_{CB}\right) & (\text{where } N_{CB} \in_R \{0,1\}^k) \\
B \to C: & (sid, [b]P, \text{MAC}_{N_{CB}}(sid, [b]P, C)) & \\
C \to A: & (sid, [c]P) & (\text{where } c \in_R \mathbb{Z}_n) \\
A \to C: & \left(sid, [c]P,\ {}^A\mathcal{E}_C(N_{AC}\right) & (\text{where } N_{AC} \in_R \{0,1\}^k) \\
C \to A: & (sid, [c]P, \text{MAC}_{N_{AC}}(sid, [c]P, A)) & \\
C \to B: & (sid, [c]P) & \\
B \to C: & \left(sid, [c]P,\ {}^B\mathcal{E}_C(N_{BC}\right) & (\text{where } N_{BC} \in_R \{0,1\}^k) \\
C \to B: & (sid, [c]P, \text{MAC}_{N_{BC}}(sid, [c]P, B)) &
\end{array}
$$

$$Key : e(P,P)^{abc} = e([b]P,[c]P)^a = e([a]P,[c]P)^b = e([a]P,[b]P)^c$$

**Protocol 14:** Joux protocol authenticated with $\lambda_{\text{ENC}}$, not optimized

$$
\begin{array}{lll}
B \to A: & (sid, [b]P) & (\text{where } b \in_R \mathbb{Z}_n) \\
A \to B: & \left(sid, [b]P, \ {}^{A}\mathcal{E}_B(N_{AB})\right) & (\text{where } N_{AB} \in_R \{0,1\}^k) \\
B \to A: & (sid, [b]P, \mathrm{MAC}_{N_{AB}}(sid, [b]P, A)) & \\
C \to A: & (sid, [c]P) & (\text{where } c \in_R \mathbb{Z}_n) \\
A \to C: & \left(sid, [c]P, \ {}^{A}\mathcal{E}_C(N_{AC})\right) & (\text{where } N_{AC} \in_R \{0,1\}^k) \\
C \to A: & (sid, [c]P, \mathrm{MAC}_{N_{AC}}(sid, [c]P, A)) & \\
A \to B: & (sid, [a]P, [c]P) & (\text{where } a \in_R \mathbb{Z}_n) \\
B \to A: & \left(sid, [a]P, [c]P, \ {}^{B}\mathcal{E}_A(N_{BA})\right) & (\text{where } N_{BA} \in_R \{0,1\}^k) \\
A \to B: & (sid, [a]P, [c]P, \mathrm{MAC}_{N_{BA}}(sid, [a]P, [c]P, B)) & \\
A \to C: & (sid, [a]P, [b]P) & \\
C \to A: & \left(sid, [a]P, [b]P, \ {}^{C}\mathcal{E}_A(N_{CA})\right) & (\text{where } N_{CA} \in_R \{0,1\}^k) \\
A \to C: & (sid, [a]P, [b]P, \mathrm{MAC}_{N_{CA}}(sid, [a]P, [b]P, C)) & \\
\multicolumn{3}{c}{Key: e(P,P)^{abc} = e([b]P, [c]P)^a = e([a]P, [c]P)^b = e([a]P, [b]P)^c}
\end{array}
$$

**Protocol 15:** Variant of Joux protocol authenticated with $\lambda_{\text{ENC}}$, not optimized

# B Proof of Theorem 1

This appendix provides the proof of Theorem 1, which is repeated below for convenience:

**Theorem 1** *Given the* DBDH *assumption, Protocols 2 and 3 are both* SK*-secure in the* AM*.*

**Proof:** The proof is almost identical to that of the Diffie-Hellman key exchange provided in [9]. However, changes are required to cater for the participation of three parties instead of two and the use of the DBDH assumption instead of the decisional Diffie-Hellman assumption. To see that the first requirement of Definition 3 is met (if three uncorrupted parties complete matching sessions then they all output the same key), note that if $A$, $B$ and $C$ are uncorrupted, then they all compute the same key. Note that the session identifier, $sid$, uniquely binds the values $[a]P$, $[b]P$ and $[c]P$ as well as the identities of the parties, $A$, $B$ and $C$, to these particular matching sessions. The values are differentiated by $sid$ from other values that $A$, $B$ and $C$ may exchange in other (possibly simultaneous) sessions. In addition, the values are differentiated by $sid$ from other values that any of $A$, $B$ or $C$ may exchange with different parties (i.e. parties other than $A$, $B$ or $C$).

To see that the second requirement of Definition 3 is met by both protocols (the probability that $\mathcal{A}$ guesses correctly the bit $b$ saying whether it received the key or a random value for the test session it chose is no more than 1/2 plus a negligible function in the security parameter), assume to the contrary there is a KE-adversary $\mathcal{A}$ in the AM against the protocol that has a non-negligible advantage in guessing correctly the bit $b$. A distinguisher $\mathcal{D}$ that distinguishes between the distributions $Q_0$ and $Q_1$ with non-negligible probability can then be constructed using $\mathcal{A}$, thus reaching a contradiction with the above DBDH assumption. The input to $\mathcal{D}$ is denoted by $(\mathbb{G}_1, \mathbb{G}_2, P, e, \alpha, \beta, \gamma, \delta)$ and is chosen from $Q_0$ or $Q_1$ each with probability 1/2. Let $l$ be an upper bound on the number of sessions invoked by $\mathcal{A}$ in any interaction. The distinguisher $\mathcal{D}$ uses adversary $\mathcal{A}$ as a subroutine and proceeds as follows:

1. Choose $r \in_R \{1 \ldots l\}$. Invoke $\mathcal{A}$ on a simulated interaction in the AM with parties $P_1, \ldots, P_m$ running the above protocol. Hand $\mathcal{A}$ the values specifying $\mathbb{G}_1, \mathbb{G}_2$ as the public parameters for the protocol execution.
2. Whenever $\mathcal{A}$ activates a party to establish a new session (except for the $r$-th session) or to receive a message, follow the instructions of the protocol on behalf of that party. When

a session is expired at a player erase the corresponding session key from that player's memory. When a party is corrupted or a session (other than the $r$-th session) is exposed, hand $\mathcal{A}$ all the information corresponding to that party or session as in a real interaction.

3. When the $r$-th session, say $(A, B, C, sid)$, is invoked to exchange a key between $A$, $B$ and $C$, let the protocol be carried out as specified, except that the values $[a]P$, $[b]P$ and $[c]P$ are replaced with $\alpha$, $\beta$ and $\gamma$.
4. If session $(A, B, C, sid)$ is chosen by $\mathcal{A}$ as the test-session, then provide $\mathcal{A}$ with $\delta$ as the answer to this query.
5. If the $r$-th session $(A, B, C, sid)$ is ever exposed, or if a session different to the $r$-th session is chosen as the test-session, or if $\mathcal{A}$ halts without choosing a test-session then $\mathcal{D}$ outputs $b' \in_R \{0, 1\}$ and halts.
6. If $\mathcal{A}$ halts and outputs a bit $b'$ then $\mathcal{D}$ halts and outputs $b'$ too.

Note that the run of $\mathcal{A}$ by $\mathcal{D}$ (up to the point where $\mathcal{A}$ stops or $\mathcal{D}$ aborts $\mathcal{A}$'s run) is identical to a normal run of $\mathcal{A}$ against the above protocol. In the case where the test session *sid* chosen by $\mathcal{A}$ coincides with the session chosen at random by $\mathcal{D}$ (i.e., the $r$-th session as chosen in Step 1), the response to the test-query of $\mathcal{A}$ is $\delta$. Thus, if the input to $\mathcal{D}$ came from $Q_0$ then the response was the actual value of the key exchanged between $A$, $B$ and $C$ during the test-session *sid* (since, by construction, the session key exchanged in Step 3 of the instructions for $\mathcal{D}$ is $\delta = e(P, P)^{abc}$). On the other hand, if the input to $\mathcal{D}$ came from $Q_1$ then the response to the test query was a random pairing, i.e. a random value from the distribution of keys generated by the protocol. In addition, the input to $\mathcal{D}$ was chosen with probability $1/2$ from $Q_0$ and with probability $1/2$ from $Q_1$ and so the distribution of responses provided by $\mathcal{D}$ to the test query of $\mathcal{A}$ is the same as specified in the definition of SK-security. In this case, the probability that $\mathcal{A}$ guesses correctly the value of $b$ is $1/2 + \epsilon$ for non-negligible $\epsilon$. However, this is equivalent to guessing whether the input to the distinguisher $\mathcal{D}$ came from $Q_0$ or $Q_1$, respectively. Thus, by outputting the same bit as $\mathcal{A}$, the distinguisher $\mathcal{D}$ guesses correctly the input distribution $Q_0$ or $Q_1$ with the same probability, $1/2 + \epsilon$, as $\mathcal{A}$ did.

Now consider the case in which the $r$-th session is not chosen as a test-session. In this case $\mathcal{D}$ always ends outputting a random bit, and thus its probability of guessing correctly the input distribution is $1/2$.

Since the first case (in which the test-session and the $r$-th session coincide) happens with probability $1/l$ while the other case happens with probability $1 - 1/l$ we find that the overall probability of $\mathcal{D}$ guessing correctly is $1/2 + \epsilon/l$, and thus $\mathcal{D}$ succeeds in distinguishing $Q_0$ from $Q_1$ with non-negligible advantage. Since this contradicts the original DBDH assumption, the assumption that there is an adversary $\mathcal{A}$ in the AM against the protocol that has a non-negligible advantage in guessing correctly the value of $b$ is false. Hence the second requirement of Definition 3 is met and this completes the proof. □