Strongly-Secure Identity-based Key Agreement and Anonymous Extension[⋆]

Sherman S.M. Chow[1] and Kim-Kwang Raymond Choo[2,⋆⋆]

[1] Department of Computer Science
Courant Institute of Mathematical Sciences
New York University, NY 10012, USA
schow@cs.nyu.edu

[2] Australian Institute of Criminology
GPO Box 2944, Canberra ACT 2601, Australia
raymond.choo@aic.gov.au

Abstract. We study the provable security of identity-based (ID-based) key agreement protocols. Although several published protocols have been proven secure in the random oracle model, only a weak adversarial model is considered – the adversary is not allowed to ask Session-Key Reveal queries that will allow the adversary to learn previously established session keys. Recent research efforts devoted to providing a stronger level of security require strong assumptions, such as assuming that the simulator has access to a non-existential computational or decisional oracle. In this work, we propose an ID-based key agreement protocol and prove its security in the widely accepted indistinguishability-based model of Canetti and Krawczyk. In our proof, the simulator does not require access to any non-existential computational or decisional oracle. We then extend our basic protocol to support ad-hoc anonymous key agreement with bilateral privacy. To the best of our knowledge, this is the first protocol of its kind as previously published protocols are for fixed group and provide only unilateral privacy (i.e., only one of the protocol participants enjoy anonymity).

Key words: Key agreement, identity-based cryptography, reveal query, provable security, anonymity

1 Introduction

To establish secure communications over an insecure communication channel, a common secret (session) key to be shared among the communicating parties is often established using key establishment protocols. Key establishment protocols can be broadly categorised into key agreement protocols or key transport protocols depending on the nature of the session key (whether input to the session key is required from only one party or all the participating parties).

The basis of many key establishment protocols relies on the Diffie–Hellman key exchange and the RSA algorithm [15, Chapter 2]. In recent years, elliptic curve cryptography has emerged as a promising branch of public-key cryptography particularly due to its potential for offering similar security to established public-key cryptosystems at reduced key sizes. We observe an emerging trend in the use of identity-based cryptography, such as a large number of identity-based key agreement protocols based on pairings. As pointed out in a recent survey article [6], the public keys in ID-based protocols are arbitrary bit-strings and can include any descriptive information including temporal information. The corresponding private key is then generated by a trusted key generation center (KGC). The strength of ID-based systems in terms of a simplified key management system (i.e., no public key certificates required) is also one of its weaknesses – since users are

[⋆] The abridged version of this paper appears in the proceedings of Information Security Conference (ISC 2007), volume 4779 of Lecture Notes in Computer Science, pages 315–332.

[⋆⋆] The views and opinions expressed in this paper are those of the author and do not reflect those of the Australian Government or the Australian Institute of Criminology. This research was not undertaken as part of the author's work at the Australian Institute of Criminology.

not allowed to generate their own private keys, key escrow is, therefore, inevitable. Key agreement protocols, however, can be used to establish session keys not under KGC's escrow.

We now highlight two other on-going research problems in the design of ID-based key agreement protocols, the focus of this paper.

Security issues: Session-Key Reveal ***and*** Session-State Reveal ***queries***
The purported security of many ID-based protocols for two parties is proven in a weak variant of Bellare–Rogaway (BR) model [3] in which the adversary is not allowed to ask any Session-Key Reveal[1] query [6]. Protocols proven secure in such a restricted model (hereafter referred to as the wBR model) do not provide the known-key security attribute, meaning that compromise of previously accepted session keys may affect the security of a non-related session.

To better explain the Session-Key Reveal (and the stronger notion of Session-State Reveal queries in the Canetti–Krawczyk model), we recall the unauthenticated Diffie-Hellman key exchange protocol as described in Figure 1. In the protocol, all arithmetic is performed modulo a large prime p with q being the prime order of g, $\in_R$ denotes choosing an element uniformly at random from the corresponding domain, $\mathcal{K}$ denotes a key derivation function (which can be realized by a hash function mapping to the secret key domain of some symmetric cryptographic scheme), and sk denotes the session key established at the conclusion of the protocol execution.

We now execute the protocol described in Figure 1 twice. Two independent sessions with the respective session keys, $sk_1 = g^{ab}$ and $sk_2 = g^{a'b'}$, where $a \neq a', b'$ and $b \neq a', b'$, were established. We assume that there exists a malicious adversary who is interested to learn the session key associated with one of the sessions, e.g., $sk_1 = \mathcal{K}(g^{ab})$ – the session key associated with the first session.

$$
\begin{array}{ccc}
\hline
a \in_R \mathbb{Z}_q & \xrightarrow{\quad g^a \quad} & b \in_R \mathbb{Z}_q \\
sk = \mathcal{K}((g^b)^a) & \xleftarrow{\quad g^b \quad} & sk = \mathcal{K}((g^a)^b) \\
\hline
\end{array}
$$

Fig. 1. Diffie–Hellman Protocol

– In a security model that allows the adversary to ask the Session-Key Reveal query, the adversary is allowed to learn session key associated with any non-related session, i.e., $sk_2 = \mathcal{K}(g^{a'b'})$.
– In a security model that allows the adversary to ask the Session-State Reveal query, the adversary is allowed to learn the ephemeral parameters of any non-related sessions. In this case, the adversary is allowed to learn either the ephemeral DH keys, a' and b', or the keying material $g^{a'b'}$, if they have not been erased from the internal state of the respective entity.

The significance of Session-State Reveal queries stems from the fact that a user may decide to store the pre-computed results to be used in future session key establishment for efficiency. These parameters, often not protected as securely as the long-term private key, may be exploited by the adversary. Such a query is designed to consider the leakage of such ephemeral parameters.

It is common practice to prove the strongest security that we can claim about any cryptographic scheme and this seems a sound principle to follow in the case of ID-based key agreement protocols. It is, therefore, not surprising that we advocate the importance of proving ID-based protocols secure in a security model that allows the adversary to ask both the Session-Key Reveal and Session-State Reveal queries. Protocols proven secure in such a model will also assure protocol implementers that they provide known-key security attribute and provide resilience against the leakage of ephemeral parameters.

[1] The Session-Key Reveal query allows the adversary to learn previously established session keys.

Privacy issues: Confidentiality of identity
Anonymity is required in many applications to ensure that the identifying information about the user is not revealed. This concept is also useful and applicable to key agreement protocol. Suppose two entities, $\mathcal{U}$ and $\mathcal{V}$, want to exchange confidential messages. In anonymous key agreement protocols such as the protocols of Boyd and Park [8] and of Shoup [32], $\mathcal{U}$'s identity is not known to anyone in the network except $\mathcal{V}$ – the recipient entity in the key agreement protocol.

This work considers anonymity from a slightly different perspective. Although $\mathcal{V}$ knows that $\mathcal{U}$ is a member of a group of users, $\mathcal{V}$ is unable to confirm the actual identity of $\mathcal{U}$. This class of protocol is useful when $\mathcal{V}$ only needs to ensure the membership of the sender, but not the identity of the user, perhaps, due to privacy issues. Our protocol provides deniability [7] for any user who has taken part in a protocol run to deny that this was the case, since any one can simulate runs of the protocol involving any other potential user.

2 Related Work and Our Contributions

2.1 Session-Key Reveal and Session-State Reveal Queries

Recent research efforts have been devoted towards designing protocols that can be proven secure in a model that allows the Session-Key Reveal queries. For example, the ID-based protocols of Chen and Kudla [10] and McCullagh and Barreto [25] were improved [18] to ensure that these protocols can be proven secure in a less restrictive sense (the adversary is allowed to ask Session-Key Reveal queries in most cases) in the random oracle model, assuming bilinear Diffie-Hellman problem is intractable. The technicality of not being able to answer reveal queries in some special sessions can be resolved using the gap assumption – the underlying computational problem is intractable even with the help of a corresponding decisional oracle.

Using the gap assumption, Kudla and Paterson [23] propose a generic transformation turning two-party Diffie–Hellman-based protocols proven secure in the wBR model to one in the full BR model. This is also applicable to two-party ID-based protocols such as the protocols of Chen and Kudla [10] and McCullagh and Barreto [25]. However, gap assumption in [10] and [25] means the simulator has access to a decisional *bilinear* Diffie-Hellman oracle (in contrast with decisional Diffie-Hellman oracle that can be realized by some classes of pairing). Chow ([19] as cited in [18]) raised a similar observation.

Along somewhat similar line, Wang [35] proposes a protocol based on a decisional problem by using a *computational oracle* to support the Session-Key Reveal queries. Again, the simulation in this proof requires the existence of a special oracle. Finally, we note that Cheng *et al.* [13] introduce the concept of coin queries that forces the adversary to reveal its ephemeral secret, and thus making Session-Key Reveal possible. Their approach is restricted in the sense that the possibility of breaking a protocol without knowing the ephemeral secret (which is possible in a real world attack) is not modelled.

The Session-State Reveal query in the Canetti–Krawczyk model (hereafter referred to as the CK model) [9] allows an adversary to learn the ephemeral parameters associated with a particular session. An example of a protocol secure in this stronger model[2] is the HMQV protocol [22], which is the "hashed" variant of the MQV protocol[3]. The basic version of HMQV is proven secure even if the adversary is allowed to ask Session-Key Reveal queries under the computational Diffie-Hellman assumption. The enhanced version of HMQV is proven secure even when the adversary learns the ephemeral Diffie–Hellman key associated with any non-target sessions, under the gap Diffie-Hellman assumption and knowledge of exponent assumption [2]. No security claim is, however, made about the availability of the keying material for the derivation of the session key.

Our contribution: High-performance ID-based key agreement protocol
We propose a new ID-based key agreement protocol. Security assurance of the protocol is provided in the

[2] The relative strengths between the BR and CK models are discussed in [17].
[3] MQV's security is analyzed [24]. without consideration of Session-State Reveal query.

stronger CK model, which allows the adversary to ask Session-Key Reveal queries in *all* cases, and Session-State Reveal queries in most cases, *without* employing any gap assumption. We show how to provide *KGC forward secrecy* by making minor modifications to the (basic) protocol. Additional parameters included in our session state definition are the ephemeral Diffie–Hellman (DH) key of the outgoing DH values and the keying material for the key derivation. Among the ID-based two-party protocols surveyed in [6], our proposed protocols achieve the strongest security properties without compromising on efficiency.

2.2 Anonymous Key Agreement Protocols

To illustrate the usefulness of our proposed key agreement protocols, we now consider the scenario of delegates making and receiving phone calls on their mobile phones while international roaming. Before secure roaming can be established, the service provider must verify whether the roaming user is a legitimate subscriber with the respective home server. Conventional anonymous roaming mechanisms [1, 29] are rather inefficient as users would have to wait online while foreign telecommunication network communicates with the original home server to authenticate the users. These geographically distributed servers also generate extra network traffic during this process. At the same time, it is inconvenient to constantly *renew* the alias in an *unlinkable* manner to hide the identities.

Our proposed key agreement protocols with the anonymity feature allow user to "hide" among a group of subscribers associated with the same home server. Moreover, after the home server has issued sets of matching public/private key pair at the very beginning, the home server is no longer required to be online. Our approach does not, however, appear to be *scalable* if one needs to hide among all (a potentially large set of) legitimate subscribers, and may not be *flexible* since it is natural that the set of subscribers is constantly changing. Both issues can be readily solved without an *a priori* group formation step. For example, any legitimate user will be able to spontaneously conscript an arbitrary group of users (i.e., without cooperation from other parties in the group) for each session. Such *ad-hoc* group formation empowers a user to have full control over the level of anonymity desired during the secure roaming establishment process.

Although alias should also be used in our approach so that the list of users can be made available to users without revealing any user information, no renewal of alias (and possibly renewal of credential) is necessary as different invocations are unlinkable (guaranteed by the unconditional anonymity of our protocol).

Our contribution: Key agreement protocol with bilateral privacy
Motivated by the various applications of anonymous roaming and our observation that existing research (e.g., see [12]) appears to focus only on *unilateral* identity privacy (i.e., only one protocol participant enjoys anonymity), we propose a secure key exchange among anonymous users in different spontaneous groups. Spontaneity and bilateral privacy features in our proposed protocol are particularly applicable in ad-hoc group communication settings. Furthermore, as noted in the literature of ID-based ring signature (e.g., [20]), ID-based solution provides a higher level of spontaneity and efficiency than conventional public key cryptosystem since one can conscript virtually anyone and no verification of public key certificates is required. With these benefits in mind, we introduce the notion of ID-based ad-hoc anonymous key agreement with *bilateral* privacy, which is realized by an extension of our basic protocol. Note that our approach is fundamentally different from that of Cheng *et al.* [12].

2.3 Challenge-Response Signatures

Exponential challenge-response (XCR) signature is introduced by Krawczyk [22] as a building block for key agreement protocol. Such a scheme, an inherently interactive public key signature protocol, requires the verifier to issue a challenge to the signer and the later responses by a signature on a given message. Apart from being unforgeable, challenge-response signature ensures that only the legitimate signer can generate a signature which will make the challenger convinced, i.e. challenge-response signature is challenge-specific. A distinctive feature of Krawczyk's XCR scheme is that any party who possesses the challenge can always generate the *same* signature string as the signer. This property of XCR, essentially, makes it the building block for key agreement protocol.

As noted by Krawczyk [22], no assumptions regarding third party's verifiability of a XCR signature is made as the verifiability is (easily) transferable.

Our contribution: How concepts in signatures help ID-based key agreement protocols. We propose a *strong* challenge-response signature scheme in which the verification requires the knowledge of the designated verifier's private key. To the best of our knowledge, our scheme is the *first* ID-based challenge-response signature scheme.

In the digital signature paradigm, unforgeability against adaptive chosen-message attack is a standard requirement – there exists a signing oracle giving arbitrary signatures of the adversary's choice. We observe that this concept is somewhat analogous to Session-Key Reveal queries in the key agreement paradigm that returns session keys of the adversary's choice when asked. Ephemeral parameters are often involved in the computation of signatures, which are also inherent in key agreement protocol.

We also note that there are many well-established signature schemes with anonymity concerns. For example, in ring signature, formalized by Rivest, Shamir, and Tauman [27, 28], absolute anonymity is provided whereby a signer can sign a message on behalf of a spontaneous group of possible signers without exposing the identity of the actual signer. It is natural to ask, "*Can we borrow concepts from digital signatures to construct key agreement protocols?*" This work provides an affirmative answer.

We built on the idea of Krawczyk [22] to propose a "better" ID-based key agreement protocol using components from our strong pairing challenge-response signature. We also propose an anonymous key agreement protocol using components from Chow *et al.*'s ID-based ring signature [20], the structure of which fits in nicely with our challenge-response signature.

3 Number Theoretic Assumptions

Let $\mathbb{G}$ be an additive group of prime order q and $\mathbb{G}_T$ be a multiplicative group also of order q. We assume the existence of an efficiently computable bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ such that

1. There is an known element $P \in \mathbb{G}$ satisfying $\hat{e}(P, P) \neq 1_{\mathbb{G}_T}$.
2. For $Q, W, Z \in \mathbb{G}$, both $\hat{e}(Q, W + Z) = \hat{e}(Q, W) \cdot \hat{e}(Q, Z)$ and $\hat{e}(Q + W, Z) = \hat{e}(Q, Z) \cdot \hat{e}(W, Z)$.

Definition 1 (Interactive Game with a BDH Challenger [12]). *Let $\mathcal{A}$ be a pair of probabilistic polynomial-time (PPT) algorithms $(\mathcal{A}_1(r_1; \ldots), \mathcal{A}_2(r_2; \ldots))$, where r_i is used by $\mathcal{A}_i$ as the random tape, that engages with a challenger in the following game. Let (P, aP, bP, cP) be the BDH instance where $P, aP, bP, cP \in \mathbb{G}$ and $a, b, c \in \mathbb{Z}_q^*$. The game is defined as follows.*

Stage 1: $(X, \sigma) \leftarrow \mathcal{A}_1(r_1; P, aP, bP, cP, \hat{e}, \mathbb{G}, \mathbb{G}_T, q)$ *(σ denotes some state)*
Interactive Part: *After seeing X, challenger returns a random $h \leftarrow_R \mathbb{Z}_q^*$.*
Stage 2: $K \leftarrow \mathcal{A}_2(r_2; h, \sigma)$.

We say that the adversary, $\mathcal{A}$, wins the game if it computes $K = \hat{e}(aP, X + hbP)^c$.

If X is determined *after* seeing h, the problem is easy since one can set $X = rP - hbP$ for $r \in_R \mathbb{Z}_q^*$, and returns $K = \hat{e}(aP, cP)^r$. It explains the game's interactive nature.

The following lemma says that if the BDH problem is hard, any adversary can only have a negligible advantage in winning the interactive BDH game. The proof is similar to the one presented in [12].

Lemma 1 (Interactive BDH Game Assumption) *For any adversary with PPT algorithm $(\mathcal{A}_1, \mathcal{A}_2)$ with advantage $\epsilon(k)$ to win the interactive BDH game, there exists an algorithm that solves BDH problem with probability $\epsilon(k)^2$.*

Proof. Given a BDH problem instance $(P, aP, bP, cP, \hat{e}, \mathbb{G}, \mathbb{G}_T, q)$, we construct a BDH solver $\mathcal{B}$ making use of $(\mathcal{A}_1, \mathcal{A}_2)$ as follows.

$\mathcal{B}$ starts by choosing two elements h and h' randomly from $\mathbb{Z}_q^*$. $\mathcal{B}$ calls $(X, \sigma) \leftarrow \mathcal{A}_1(r_1; P, aP, bP, cP, \hat{e}, \mathbb{G}, \mathbb{G}_T, q)$ and $K \leftarrow \mathcal{A}_2(r_2; h, \sigma)$. $\mathcal{B}$ now rewinds the adversary backs to the point before $\mathcal{A}_2$ is called. $\mathcal{A}_2$ is then executed again with h' to get $K \leftarrow \mathcal{A}_2(r_2; h', \sigma)$. Since h and h' are chosen independently from X and σ, the probability each of two executions of $\mathcal{A}_2$ returns a valid answer is at least $\epsilon(k)$. Under such condition, $K = \hat{e}(aP, X + hbP)^c$ and $K' = \hat{e}(aP, X + h'bP)^c$. Ignoring the negligible probability that $h = h'$, $\hat{e}(aP, bP)^c$ can be obtained by $(K/K')^{(h-h')}$, i.e., $\mathcal{B}$ solves BDH problem with probability $\epsilon(k)^2$. □

In the proofs of Kudla and Paterson [23] and Wang [35], the required (decisional BDH) oracle, in which the simulator has access to, has no known polynomial time realization. Their assumptions are non-falsifiable whilst in our case, we only assume the BDH problem is intractable, something that can be falsified.

We also consider a variant of the BDH problem, the Modified Bilinear Diffie-Hellman(MBDH) problem, for the proof of our escrow-free protocol.

Definition 2 (Modified (Computational) Bilinear Diffie-Hellman (MBDH) Problem [21]). *Given $(P, aP, bP, cP, c^{-1}P)$, output $\hat{e}(P, P)^{abc} \in G_2$.*

Computational and decisional MBDH problems were first proposed in [21] to realize the first ID-based signcryption scheme with forward-secrecy and public ciphertext authenticity. In this paper, we reduce the security of our escrow-free protocol to an interactive MBDH assumption, which is defined in a way similar to Definition 1 (adding an extra element to be supplied to the adversary) to support KGC forward-secrecy. We have the following result as described by Lemma 2.

Lemma 2 (Interactive MBDH Game Assumption) *For any adversary with PPT algorithm $(\mathcal{A}_1, \mathcal{A}_2)$ with advantage $\epsilon(k)$ to win the interactive MBDH game, there exists an algorithm that solves MBDH problem with probability $\epsilon(k)^2$.*

4 Strong Challenge-Response Signatures

We now describe an underlying building block of our protocol – pairing challenge-response signatures (the signatures are in $\mathbb{G}_T$, the range of the pairing function).

4.1 Deterministic Identity-Based Key Generation

Setup : On input a security parameter k, KGC uses a BDH instance generator to generate $(\mathbb{G}, \mathbb{G}_T, \hat{e})$ where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order q and $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is the pairing function. KGC also chooses two cryptographic hash functions $\mathcal{H} : \{0,1\}^n \rightarrow \mathbb{G}$ and $\mathcal{H}_0 : \mathbb{G} \times \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ and a key derivation function $\mathcal{K}$. All three of these are modelled as random oracles. Then KGC randomly chooses an arbitrary generator P of $\mathbb{G}$. An element s is randomly chosen from $\mathbb{Z}_q^*$ as the KGC's master secret, and the corresponding public key is $P_{pub} = sP$. Finally, the set of public parameters is published as params $= \langle \mathbb{G}, \mathbb{G}_T, q, \hat{e}, P, P_{pub}, \mathcal{H}, \mathcal{H}_0, \mathcal{K} \rangle$.

Extract ([5]): On input an identity ID_A and a master secret s, the public key Q_A (for A) is set as $\mathcal{H}(\mathsf{ID}_A)$, and the corresponding private key S_A is $s\mathcal{H}(\mathsf{ID}_A)$.

4.2 Strong Pairing Challenge-Response Signature Scheme

Sign: Let B be the verifier requesting for a signature on a message, m, from the signer, A.

1. B picks b randomly from $\mathbb{Z}_q^*$ and sends $(W_B, W'_B) = (bQ_B, b^{-1}Q_A)$ to A.
2. After verifying $\hat{e}(W_B, W'_B) = \hat{e}(Q_B, Q_A)$, A picks $a \in_R \mathbb{Z}_q^*$ and sends $W_A = aQ_A$ and $e_A = \hat{e}((a + h_A)S_A, W_B)$ to B where $h_A = \mathcal{H}_0(W_A, m)$.

Verify: On input a signature $(W_A, e_A) \in \mathbb{G} \times \mathbb{G}_T$, a message $m \in \{0,1\}^*$, a challenge $b \in \mathbb{Z}_q^*$, a *private key* S_B and an identity string ID_A output accept if $\hat{e}(bS_B, W_A + h_A Q_A) = e_A$ where $h_A = \mathcal{H}_0(W_A, m)$, else output reject.

Discussion: Notice that bQ_B is used as the challenge instead of bP. If bP is used, B can thus easily share the verifiability of the signature with another party C as follows.

1. B prepares a challenge $W_B = bP$, then B forwards it as the challenge in the above protocol to get a signature from A.
2. Once the signature is obtained, both B and C can then verify the signature by sharing the knowledge of b.

Such an attack is possible since leaking b does no harm to B (i.e., B's private key is not compromised). We remark that Krawczyk's XCR scheme [22] suffers from a similar problem although this is not the original goal of Krawczyk to ensure non-transferability. Non-transferability of our scheme is ensured by requiring the knowledge of the private key in the verification.

Moreover, the bilinearity ensures that the challenge is in the correct form (assuming the discrete logarithm of Q_B with respect to Q_A is unknown). If symmetric pairing is used, it is possible that $(W_B, W'_B) = (b^{-1}Q_A, bQ_B)$ can pass the test; but B is unable to verify the signature if W_B is computed in this way.

4.3 Dual Challenge-Response Signature Scheme

A dual version of XCR scheme is proposed in [22], which means both parties assume the dual roles of challenger and signer; and each of them will produce a signature that no third party can forge.

With the notation introduced previously, our ID-based dual challenge-response signature on message m_A and m_B is in the form of $(W_A = aQ_A, W_B = bQ_B, e = \hat{e}(Q_A, Q_B)^{s(a+h_A)(b+h_B)})$, where $h_A = \mathcal{H}_0(W_A, m_A)$ and $h_B = \mathcal{H}_0(W_B, m_B)$. Both parties can compute e by either $\hat{e}((a + h_a)S_A, W_B + h_B Q_B)$ or $\hat{e}(W_A + h_A Q_A, (b + h_B)S_B)$.

Like the XCR signature, the verifier who chooses the challenge in the protocol can compute exactly the same pairing challenge-response signature using the random value chosen, which is the essential property for us to derive the key agreement protocol. Proofs for both schemes can be found in Appendix B.

5 High Performance ID-based Key Agreement Protocol

5.1 Basic Construction

Our proposed high performance identity-based key agreement protocol is described in Figure 2. The notation used in the protocol is as follows: (Q_U, S_U) denotes the public/private key pair for protocol participant U, sk_U and sid_U denote the session key and session identifier for protocol participant U respectively and $||$ denotes the concatenation of messages.

5.2 Security Evaluation: An Overview

The simulator, $\mathcal{S}$, knows how to answer all but one Corrupt queries, ID_J. The hard problem will be embedded in one of the sessions having ID_J as the responder. Note that neither the Session-Key Reveal queries nor the Session-State Reveal queries are allowed for this test session. For all other sessions having ID_J as the responder, $\mathcal{S}$ can correctly answer the queries asked since all state information and the private key of the initiator ID_I are known to $\mathcal{S}$.

The tricky part is answering queries directed at the sessions where ID_J acts as the initiator. $\mathcal{S}$ can, however, faithfully simulated the protocol execution by defining W_J before the output of the corresponding random oracle query $\mathcal{H}_0(W_J, \mathsf{ID}_K)$ is defined. $\mathcal{S}$ can then compute the session key in some way different from the protocol specification to answer the Session-Key Reveal query. As an abnormal way is used, answering the Session-State Reveal query correctly is not possible and this is our only restriction on simulating the Session-State Reveal queries.

$$\begin{array}{ccc}
\text{Initiator } A & & \text{Responder } B \\
a \in_R \mathbb{Z}_q^*; W_A := aQ_A & \xrightarrow{ID_A, W_A} & b \in_R \mathbb{Z}_q^*; W_B := bQ_B \\
h_A := \mathcal{H}_0(W_A, \mathsf{ID}_B) & & h_A := \mathcal{H}_0(W_A, \mathsf{ID}_B) \\
 & & h_B := \mathcal{H}_0(W_B, \mathsf{ID}_A) \\
 & & sk_B := \mathcal{K}(\hat{e}(W_A + h_A Q_A, (b + h_B)S_B)) \\
sid_A := ID_A || W_A || ID_B || W_B & \xleftarrow{ID_B, W_B} & sid_B := ID_A || W_A || ID_B || W_B \\
h_B := \mathcal{H}_0(W_B, \mathsf{ID}_A) & & \\
sk_A := \mathcal{K}(\hat{e}((a + h_A)S_A, W_B + h_B Q_B)) & & \\
\multicolumn{3}{c}{sk_A = \mathcal{K}(\hat{e}(Q_A, Q_B)^{s(a+h_A)(b+h_B)}) = sk_B}
\end{array}$$

Fig. 2. Proposed high-performance identity-based key agreement protocol

Theorem 1 *The protocol described in Figure 2 is secure (in the sense of Definition 6 in Appendix A) assuming that the BDH problem is hard* [4] *and $\mathcal{H}$, $\mathcal{H}_0$, and $\mathcal{K}$ are modelled as random oracles.*

Proof. Assuming that there exists an adversary $\mathcal{A}$ with a non-negligible advantage against our protocol described in Figure 2, we construct a simulator, $\mathcal{S}$, against the interactive game with a BDH challenger (the BDH problem instance is (P, xP, yP, zP) and the last part of the challenge is h), using $\mathcal{A}$ as a subroutine. $\mathcal{S}$ now simulates the view of $\mathcal{A}$ by answering the following queries of $\mathcal{A}$.

Setup : xP is assigned to be the public key of the KGC.

$\mathcal{H}$ queries : If an $\mathcal{H}$ query is previously asked, then the stored answer in the list $L_{\mathcal{H}}$ will be returned. Denote the I^{th} distinct $\mathcal{H}$ query by ID_I. For ID_J, $\mathcal{S}$ responses with yP; otherwise, $\mathcal{S}$ chooses $r_i \in_R \mathbb{Z}_q^*$, stores it in the list $L_{\mathcal{H}}$ along with ID_I, and outputs $r_i P$.

$\mathcal{H}_0$ queries : $\mathcal{S}$ maintains a list $L_{\mathcal{H}_0}$ to ensure that previously asked queries would receive the same answer. However, special value may be plugged into the list in the simulation of the Send queries with ID_J as the initiator and ID_K as the responder.

$\mathcal{K}$ queries : $\mathcal{S}$ just needs to ensure the random oracle property of $\mathcal{K}$, by maintaining a list $L_{\mathcal{K}}$ to ensure that previously asked queries will receive the same answer. It can be seen from the rest of the proof that the simulator knows the keying materials for all sessions, while the test session is the only exception.

Corrupt *queries* : The simulation fails (event I) if the request is ID_J, otherwise the corresponding r_i is retrieved from the list $L_{\mathcal{H}}$ and $r_i(xP)$ is returned.

Send *queries (ID_I as initiator and ID_J as responder)* : Since $\mathcal{S}$ can compute the private key of ID_I so the simulation can be done as a typical protocol invocation. Except for the following special handling for the N^{th} invocation, τ is chosen randomly from $\mathbb{Z}_q^*$ and $W_{I,N} = r_I \tau(zP)$ is returned. After $W_{J,N}$ is obtained, if $(W_{J,N}, \mathsf{ID}_I)$ can be found in list $L_{\mathcal{H}_0}$, the simulation fails (event II). Otherwise, $\mathcal{S}$ dumps all maintained lists and system parameters to the tape σ, then outputs (X, σ) where $X = W_{J,N}$. The interactive BDH challenger returns $h \in_R \mathbb{Z}_q^*$. $\mathcal{S}$ reconstructs all the lists and system parameters from σ, and set $\mathcal{H}_0(W_{J,N}, \mathsf{ID}_I) = h$, which is also denoted as $h_{J,N}$.

[4] Recall that in the result of Lemma 1, we assume the BDH problem is hard. By doing so, we also assume a negligible advantage in the interactive BDH game.

Send *queries (*ID_J *as initiator and* ID_K *as responder)* : In this case, $\mathcal{S}$ knows neither the private key of the initiator ID_J, nor the ephemeral Diffie-Hellman key of the responder ID_K. However, $\mathcal{S}$ can still do a faithful simulation by manipulating the random oracle. Suppose it is the ℓ^{th} invocation of the protocol initiated by ID_J and responded with ID_K. $\mathcal{S}$ selects $\alpha_\ell, h_{J,\ell} \in_R \mathbb{Z}_q^*$, responses with $W_{J,\ell} = \alpha_\ell P - h_{J,\ell} Q_J$, and stores h_J as the response of $\mathcal{H}_0$ corresponding to the query $(W_{J,\ell}, \mathsf{ID}_K)$. α_ℓ is also stored in the auxiliary list corresponding to the $\Pi^n_{J,K}$ session.

Session-Key Reveal *queries* : For session having ID_I as initiator and ID_J as responder, and if this is not the N^{th} invocation, $\mathcal{S}$ simply uses the private key of ID_I to answer the query asked by $\mathcal{A}$ since $\mathcal{S}$ knows the ephemeral Diffie-Hellman key chosen; otherwise, it fails (event III). For the case (ID_J, ID_K), suppose $h_{J,\ell} = \mathcal{H}_0(W_{J,\ell}, \mathsf{ID}_K)$ and $h_{K,\ell} = \mathcal{H}_0(W_{K,\ell}, \mathsf{ID}_J)$. $\mathcal{S}$ retrieves α_ℓ and returns $\mathcal{K}(\hat{e}(\alpha_\ell(xP), W_{K,\ell} + h_{K,\ell}Q_K))$. Consistency can be easily seen:

$$\begin{aligned}\mathcal{K}(\hat{e}(\alpha_\ell(xP), W_{K,\ell} + h_{K,\ell}Q_K)) &= \mathcal{K}(\hat{e}(\alpha_\ell P, W_{K,\ell} + h_{K,\ell}Q_K)^x)\\ &= \mathcal{K}(\hat{e}(\alpha_\ell P - h_{J,\ell}Q_J + h_{J,\ell}Q_J, W_{K,\ell} + h_{K,\ell}Q_K)^x)\\ &= \mathcal{K}(\hat{e}(W_{J,\ell} + h_{J,\ell}Q_J, W_{K,\ell} + h_{K,\ell}Q_K)^x).\end{aligned}$$

Session-State Reveal *queries* : For session having ID_I as initiator and ID_J as responder, it is trivial to obtain the ephemeral Diffie-Hellman key, except for the N^{th} invocation where $\mathcal{S}$ will fail (event IV). For (ID_J, ID_K), it is not supported.

$\mathcal{S}$ knows all the outgoing and incoming DH values, even for the N^{th} invocation between ID_I and ID_J and invocations between ID_J and arbitrary ID_K. $\mathcal{S}$ also knows the keying material for all sessions, except the N^{th} invocation (event IV).

Test *queries* : Suppose $h_{I,N} = \mathcal{H}_0(W_{I,N}, \mathsf{ID}_J)$ and $h_{J,N} = \mathcal{H}_0(W_{J,N}, \mathsf{ID}_I) = h$. If $\mathcal{A}$ does not choose the session $\Pi^N_{I,J}$, $\mathcal{S}$ aborts (event V). $\Pi^N_{I,J}$ should hold a session key of the following form.

$$\begin{aligned}&\mathcal{K}(\hat{e}(W_{I,N} + h_{I,N}Q_I, W_{J,N} + h_{J,N}Q_J)^x) = \mathcal{K}(\hat{e}(r_I\alpha(zP) + h_{I,N}r_IP, X + hyP)^x)\\ &= \mathcal{K}(\hat{e}((\alpha z + h_{I,N})r_IP, X + hyP)^x) = \mathcal{K}(\hat{e}(xP, X + hyP)^{(\alpha z + h_{I,N})r_I}).\end{aligned}$$

$\mathcal{S}$ cannot compute $\mathcal{K}(\hat{e}(xP, X + h(yP))^{z(r_I\alpha)})$ by itself without the assistance of $\mathcal{A}$. Therefore, $\mathcal{S}$ is unable to return the real session key. A random key drawn from session key distribution (range of $\mathcal{K}$) will be returned instead.

Answering interactive BDH challenger : If $\mathcal{S}$ does not abort and $\mathcal{A}$ is able to distinguish between real session key and random session key (with probability $\epsilon(k)$), then $\mathcal{A}$ must have queried the key derivation oracle $\mathcal{K}$ for the keying material $\hat{e}(xP, X + hyP)^{(\alpha z + h_{I,N})r_I} = \hat{e}(xP, X + h(yP))^{z(r_I\alpha)}\hat{e}(xP, X + h(yP))^{h_{I,N}r_I}$ (we ignore the small probability that $\mathcal{A}$ correctly guess this value without making the corresponding $\mathcal{K}$ query – a standard argument in random oracle model). Now $\mathcal{S}$ randomly chooses one of $\mathcal{A}$'s $\mathcal{K}$'s queries π. If $\mathcal{S}$ is lucky enough that π is the above keying material (event VI), $\mathcal{S}$ answers the interactive BDH challenger correctly with $(\pi/(\hat{e}(xP, X + h(yP))^{h_{I,N}r_I})^{1/(r_I\alpha)}$.

Probability analysis :

I. If event V does not occur, neither does event I.
II. Let $N_{\mathcal{H}}$ be the number of $\mathcal{H}_0$ queries and k be the security parameter, collusion would not occur with probability $(2^k - N_{\mathcal{H}})/2^k$.
III. If event V does not occur, neither does event III.
IV. If event V does not occur, neither does event IV.
V. Let $N_{\mathcal{C}}$ be the number of sessions created, $\mathcal{A}$ chooses the session $\Pi^N_{I,J}$ with probability $1/N_{\mathcal{C}}$.
VI. Let $N_{\mathcal{K}}$ be the number of key derivation oracle queries, event VI occurs with probability $1/N_{\mathcal{K}}$.

$\mathcal{S}$ wins the game if event II and V does not occur but event VI occurs. If $\mathcal{A}$ is able to have an advantage $\epsilon(k)$ against our protocol, then $\mathcal{S}$ can also win with an advantage of at least $\frac{\epsilon(k)(2^k - N_{\mathcal{H}})}{N_{\mathcal{C}} N_{\mathcal{K}} 2^k}$. However, since such an adversary $\mathcal{A}$ does not exist, the proof for Theorem 1 follows easily. □

Key compromise may lead to another problem. When the long-term key of an entity, A, is compromised; the adversary may be able to masquerade not only *as* A but also *to* A as another party, B. Our protocol is resistance to such attacks.

Theorem 2 *The protocol described in Figure 2 provides key compromise impersonation resilience (*KCIR*) assuming that the BDH problem is hard and $\mathcal{H}$, $\mathcal{H}_0$, and $\mathcal{K}$ are modelled as random oracles.*

Proof. Following the approaches of Chen and Kudla [10] and Krawczyk [22], we make a slight modification to the security model to capture KCIR – $\mathcal{A}$ is allowed to corrupt the initiating party, ID_I. The simulation by $\mathcal{S}$ will not abort even if $\mathcal{A}$ requested for the private key of ID_I. Therefore, the proof for Theorem 1 will not be invalidated by this change and Theorem 2 follows. □

5.3 Forward-Secrecy and Escrow-Freeness

Although an adversary can masquerade as the compromised entity once the latter's long-term key has been compromised, we do not want the adversary to also obtain previously accepted session keys. Protocols that prevent this are said to provide forward secrecy. As there is usually a computational cost in providing perfect forward secrecy, it is sometimes sacrificed and a weaker notion is considered. One example is *partial* forward secrecy whereby the compromise of *one* long-term private key or *both* ephemeral secrets of the communicating parties does not lead to the leakage of previously accepted session keys. No such protection is made when *both* parties' long-term keys are compromised. This notion is considered in existing ID-based protocols such as those of Chen and Kudla [10]. For our basic protocol, the proof of indistinguishability allows the adversary to ask Corrupt query for the ID_I associated with the test session, it follows that our protocol also achieve partial forward-secrecy.

There is an additional concern in forward secrecy for ID-based protocols when compared with those in conventional public key cryptography – the master secret of the KGC is another secret that can be compromised. When this happens, the long-term keys of all users will be compromised although it may be possible that no previously accepted session keys are deduced. Achieving this notion also mean that the key agreement protocol is *escrow-free*, assuming that there is no active attack by the KGC (e.g., by actively impersonating a user). A protocol is said to provide *KGC forward secrecy* (KGC-FS) if it retains confidentiality of previously accepted session keys even when the master secret of the KGC is compromised. It is easy to see that our protocol described in Figure 2 does not provide KGC-FS since any adversary with the knowledge of s will be able to compute $\hat{e}(W_A + h_A Q_A, W_B + h_B Q_B)^s = \hat{e}(Q_A, Q_B)^{s(a+h_A)(b+h_B)}$.

KGC-FS implies forward secrecy in the usual sense since all users' private keys can be computed with the master secret. It has been noted that two-party protocols with only two-message flow and having no previous establishment of secure shared state cannot achieve *perfect* forward secrecy [22]. Our protocol, having only two messages in the message flow, inherently can neither achieve *perfect* forward secrecy nor perfect KGC-FS. Here we consider *weak* KGC-FS, such that the previously established sessions without the active involvement of the adversary cannot be "recovered" even if the long-term key is compromised. We adopt the approach of Chen and Kudla [10] to ensure that our protocol offers the same level of KGC-forward-secrecy as their protocol. The new protocol is described in Figure 3, with the underlined values indicate the changes from the basic protocol described in Figure 2.

Informally, the protocol described in Figure 3 provides KGC-FS at the expense of two additional offline scalar-point multiplications and one online scalar-point multiplication. Learning s will not help the adversary in computing $\mathcal{K}(\hat{e}((a + h_A)Q_A, (b + h_B)Q_B)^s, \underline{abP})$ as finding abP means the CDH problem is solvable (since both a and b are deleted from the internal states upon completion of the protocol execution). Using the same exponent in the elements T and W allows a saving of one pseudorandom number generation and hence, faster exponentiation operation using the same exponent is possible. Security assurance is given by the following three theorems.

$$
\begin{array}{ccc}
\text{Initiator } A & & \text{Responder } B \\
a \in_R \mathbb{Z}_q^*; W_A := aQ_A; \underline{T_A := aP} & \xrightarrow{ID_A, W_A, \underline{T_A}} & b \in_R \mathbb{Z}_q^*; W_B := bQ_B; \underline{T_B := bP} \\
h_A := \mathcal{H}_0(W_A, \underline{T_A}, \mathsf{ID}_B) & & h_A := \mathcal{H}_0(W_A, \underline{T_A}, \mathsf{ID}_B) \\
 & & h_B := \mathcal{H}_0(W_B, \underline{T_B}, \mathsf{ID}_A) \\
 & & sk_B := \mathcal{K}(\hat{e}(W_A + h_A Q_A, (b + h_B)S_B), \underline{bT_A}) \\
sid_A := \mathsf{ID}_A || W_A || \underline{T_A} || \mathsf{ID}_B || W_B || \underline{T_B} & \xleftarrow{ID_B, W_B, \underline{T_B}} & sid_B := \mathsf{ID}_A || W_A || \underline{T_A} || \mathsf{ID}_B || W_B || \underline{T_B} \\
h_B := \mathcal{H}_0(W_B, \underline{T_B}, \mathsf{ID}_A) & & \\
sk_A := \mathcal{K}(\hat{e}((a + h_A)S_A, W_B + h_B Q_B), \underline{aT_B}) & & \\
\multicolumn{3}{c}{sk_A = \mathcal{K}(\hat{e}(Q_A, Q_B)^{s(a+h_A)(b+h_B)}, \underline{abP}) = sk_B}
\end{array}
$$

Fig. 3. Proposed escrow-free high-performance identity-based key agreement protocol

Theorem 3 *The protocol described in Figure 3 is secure (in the sense of Definition 6 in Appendix A) assuming that the Modified (Computational) Bilinear Diffie-Hellman (MBDH) problem is hard and $\mathcal{H}$, $\mathcal{H}_0$, and $\mathcal{K}$ are modelled as random oracles.*

Theorem 4 *The protocol described in Figure 3 provides weak* KGC-forward-secrecy *(*KGC-FS*) assuming that the Computational Diffie-Hellman (CDH) problem is hard and $\mathcal{H}$, $\mathcal{H}_0$, and $\mathcal{K}$ are modelled as random oracles.*

Theorem 5 *The protocol described in Figure 3 provides key compromise impersonation resistance assuming that the Modified Bilinear Diffie-Hellman (MBDH) problem is hard and $\mathcal{H}$, $\mathcal{H}_0$, and $\mathcal{K}$ are modelled as random oracles.*

Proofs for Theorems 3 to 5 are presented in the Appendix C.

5.4 Comparison with Existing Protocols

Table 1 describes the summary of comparison between several two-party ID-based protocols with two message flows. M denotes scalar-point multiplication, H denotes MapToPoint function [5] hashing identity to a point on an elliptic curve, and P denotes pairing. Off-line computation can be pre-computed before the execution of the protocol, which includes public key derivation. Note that pairings are expensive and should be avoided whenever possible. MapToPoint is slightly more expensive but its cost is still comparable with that of scalar-point multiplication.

The notation wBR denotes a restricted variant of the BR model whereby Session-Key Reveal query is not supported, FS denotes user *forward secrecy* while wKGC denotes *weak KGC forward secrecy*, and KCIR denotes key compromise impersonation resistance.

As shown in Table 1, among the "unbroken" ID-based protocols that provide:

KCIR and FS (not KGC-FS). Our protocol described in Figure 2 and Wang's protocol [35] are the most efficient. However, our protocol is based on a milder assumption and yet proven secure in a stronger model, which makes it more attractive than that of Wang's.

KCIR and KGC-FS. Although our protocol described in Figure 3 is a bit less efficient as that of Chen and Kudla [10] protocol #2', our protocol is proven secure in a stronger model (allowing the adversary to ask the Session-State Reveal query).

[5] No formal proof is given, it is unclear that whether the protocol can achieve anything stronger than weak KGC-FS.

[6] It is secure in the wBR model if the mistakes in its proof are corrected.[11, 18].

Protocol	Computation			Forward Secrecy	KCIR	Proof/ Attack
	On-line	Off-line	Public Key			
Our protocol #1	$1M + 1P$	$2M$	$1H$	FS	Yes	CK
Our protocol #2	$2M + 1P$	$3M$	$1H$	wKGC	Yes	CK
Wang [35]	$2M + 1P$	$1M$	$1H$	FS	Yes	BR
The following protocols are proven secure in a restricted model.						
Chen-Kudla #2 [10]	$1P$	$2M$	$1H$	No	Yes	wBR
Chen-Kudla #2' [10]	$1M + 1P$	$3M$	$1H$	wKGC	No	wBR
McCullagh-Barreto #1 [25]	$1P$	$2M$	$1M$	FS	No	wBR
McCullagh-Barreto #2 [25]	$1P$	$2M$	$1M$	? [5]	No	wBR[6]
The following protocols do not have any security proofs.						
Smart [33]	$1P$	$2M + 1P$	$1H$	No	Yes	No
Chen-Kudla #1' [10]	$1M + 1P$	$2M + 1P$	$1H$	wKGC	Yes	No
The following protocols are broken.						
Yi [37]	$1M + 1P$	$2M$	$1H$	See Appendix E		
Choie *et al.* #1 [14]	$1M + 2P$	$2M$	$1H$	See [6]		
Choie *et al.* #2 [14]	$2M + 1P$	$2M + 1P$	$1H$	See [6]		
Shim [30]	$1P$	$2M$	$1H$	See [34]		
Xie #1 [36]	$1P$	$3M$	$1M$	See [31]		
Xie #2 [36]	$1P$	$3M$	$1M$	See [31]		

Table 1. Security and efficiency for two-party, two-message ID-based protocols

6 Ad-Hoc Anonymous Key Agreement Protocols

This section describes our extended protocol for ad-hoc anonymous key agreement based on the ID-based ring signature scheme of Chow *et al.* [20].

6.1 Our Extension

In an ad-hoc anonymous key agreement protocol, the initiator conscripts a set of users – the *initiating ring* – and similarly the responder hides in a *responding ring*. Let A_j be a member of the initiating ring $A = \{A_1, A_2, \ldots, A_J\}$ and B_k be a member of the responding ring $B = \{B_1, B_2, \ldots, B_K\}$. Note that J can be different from K. For the security proof, we require each user to derive a value ψ in each session that is different from the values chosen in previous sessions with overwhelming probability. The values of A_j and B_k are denoted by ψ_A and ψ_B respectively. Canetti and Krawczyk suggested such a pair of (ψ_A, ψ_B) constitutes a unique session identifier for each session in practice[7].

1. A_j chooses $U_i \in_R \mathbb{G}$ and computes $h_i = \mathcal{H}_0(U_i, B, \psi_A), \forall i \in \{1, \ldots, J\} \setminus \{j\}$.
2. B_k then picks $V_i \in_R \mathbb{G}$, computes $c_i = \mathcal{H}_0(V_i, A, \psi_B), \forall i \in \{1, \ldots, K\} \setminus \{k\}$.
3. A_j chooses $r'_j \in_R \mathbb{Z}_q^*$, computes $U_j = r'_j Q_{A_j} - \sum_{i \neq j}\{U_i + h_i Q_{A_i}\}$.
4. Similarly, B_k chooses $r'_j \in_R \mathbb{Z}_q^*$, computes $V_k = r'_k Q_{B_k} - \sum_{i \neq k}\{V_i + c_i Q_{B_i}\}$.
5. A_j and B_k exchange $\bigcup_{i \in \{1,\ldots,J\}} \{U_i\}$ and $\bigcup_{i \in \{1,\ldots,K\}} \{V_i\}$
6. A_j and B_k compute session key sk_A and sk_B respectively as in (♠) and (♡).

$$\begin{aligned} sk_A &= \mathcal{K}(\hat{e}((r'_j + h_j)S_{A_j}, \sum_{i=1}^{K}(V_i + c_i Q_{B_i}))) \cdots (\spadesuit) \\ &= \mathcal{K}(\hat{e}(r'_j Q_{A_j} + h_j Q_{A_j}, \sum_{i=1}^{K}(V_i + c_i Q_{B_i}))^s) \end{aligned}$$

[7] See [16, 18] for a detailed discussion on session identifiers in key establishment protocols.

$$
\begin{aligned}
&= \mathcal{K}(\hat{e}(U_j + \sum_{i \neq j}\{U_i + h_i Q_{A_i}\} + h_j Q_{A_j}, \sum_{i=1}^{K}(V_i + c_i Q_{B_i}))^s) \\
&= \mathcal{K}(\hat{e}(\sum_{i=1}^{J}(U_i + h_i Q_{A_i}), \sum_{i=1}^{K}(V_i + c_i Q_{B_i}))^s) \\
&= \mathcal{K}(\hat{e}(\sum_{i=1}^{J}(U_i + h_i Q_{A_i}), V_k + \sum_{i \neq k}\{V_i + c_i Q_{B_i}\} + c_k Q_{B_k})^s) \\
&= \mathcal{K}(\hat{e}(\sum_{i=1}^{J}(U_i + h_i Q_{A_i}), (r'_k + c_k) S_{B_k})) \\
&= \mathcal{K}(\hat{e}(\sum_{i=1}^{J}(U_i + h_i Q_{A_i}), r'_k Q_{B_k} + c_k Q_{B_k})^s) = sk_B \cdots (\heartsuit)
\end{aligned}
$$

6.2 Security Attributes

For simplicity, we assume both rings are of the same size, n. Apart from the conventional security properties for key agreement protocols, the security of ad-hoc anonymous key agreement protocols also depend on 1-out-of-n anonymity as described in Definition 3. These properties can be seen as a natural extension from the security requirements of key agreement protocol and those of ring signatures (e.g., see [20]).

Definition 3 (Security Attributes of Ad-Hoc Anonymous Key Agreement Protocols). *An ad-hoc anonymous key agreement protocol is secure if below conditions are satisfied.*

1: Validity. *If two uncorrupted oracles complete matching sessions, then both oracles must hold the same session key.*

2: Indistinguishability. *For all probabilistic, polynomial time adversaries, $\mathcal{A}$, the advantage of $\mathcal{A}$, $\mathsf{Adv}^{\mathcal{A}}(k)$, in game $\mathcal{G}$*[8] *is negligible. In particular, this implies* **1-out-of-n authenticity***: for all probabilistic, polynomial time adversaries, $\mathcal{A}$, without any one of the n private keys, has negligible advantage in learning about a fresh session key.*

3: 1-out-of-n Anonymity. *An ad-hoc anonymous key agreement protocol is said to have unconditional anonymity if for any group of n users, any adversary $\mathcal{A}$ (including the responder and the KGC) is unable to identify the real initiator better than a random guess, i.e., $\mathcal{A}$ can guess the identity of the initiator correctly with probability no better than $\frac{1}{n}$, or $\frac{1}{n-1}$ if $\mathcal{A}$ is in the ring. If the protocol satisfies* bilateral *privacy, the same requirement applies on the responding party.*

It is straightforward to see that our proposed protocol is valid. The indistinguishability and the 1-out-of-n anonymity properties are formally captured by Theorems 6 and 7 respectively. The proofs can be found in Appendix D.

Theorem 6 *The protocol described in Section 6.1 achieves indistinguishability (in the sense of Definition 3) assuming that the Bilinear Diffie-Hellman (BDH) problem is hard and $\mathcal{H}$, $\mathcal{H}_0$, and $\mathcal{K}$ are modelled as random oracles.*

Theorem 7 *The protocol described in Section 6.1 provides 1-out-of-n anonymity unconditionally.*

We remark that it is also possible to equip this protocol with weak KGC forward secrecy by using the technique presented in Section 5.3. Previously used ephemeral parameters should not, however, be re-used for full-protection of the anonymity – since the element corresponding to the real identity (U_j or V_k) should be different even all the other random factors (U_i or V_i), establishing a key reusing the random factors chosen excludes some possibilities for the real identity.

[8] Definition can be found in Appendix A.

7 Conclusion and Future Work

In conclusion, we had proposed a new identity-based (ID-based) key agreement protocol, proven secure in the Canetti–Krawczyk model that allows the adversary access to the Session-Key Reveal and Session-State Reveal queries. Our protocol is the *first* to be proven secure against such a strong adversary *without* employing any gap assumption. Using the approach of Chen and Kudla [10], we show how to provide KGC forward secrecy for our proposed ID-based protocol. Both our proposed protocols are efficient and, yet, proven secure in the strongest model among other previously published two-party two-message ID-based protocols with similar security attributes claim.

Motivated by the need for a better anonymous roaming mechanism and our observation that existing research appears to focus only on *unilateral* identity privacy, our basic protocol is extended to realize the first ad-hoc anonymous ID-based key agreement protocol with *bilateral* privacy.

Directions for future work include the following:

1. Our protocols only support the Session-State Reveal queries partially under the BDH assumption. We have seen examples of achieving a higher level of security by employing gap assumptions. For example, the security proof of the Diffie–Hellman-based HMQV protocol is strengthened when the underlying assumption is changed from computational Diffie-Hellman assumption to its gap version [22]. It will be interesting to check if our protocol can also be strengthened by using the gap BDH assumption.
2. Finding more real-world applications for our proposed ID-based ad-hoc anonymous key agreement protocol.

Acknowledgements

We would like to thank Professor Colin Boyd (Queensland University of Technology) for suggesting the problem, Shengbao Wang (Shanghai Jiao Tong University) for verifying the comparison part of the paper, and the anonymous reviewers for their invaluable feedback. Despite their invaluable assistance, any errors remaining are solely attributed to the authors.

References

1. G. Ateniese, A. Herzberg, H. Krawczyk, and G. Tsudik. Untraceable Mobility or How to Travel Incognito. *Computer Networks*, 31(8):871–884, 1999.
2. M. Bellare and A. Palacio. The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 273–289. Springer, 2004. Extended version available from http://eprint.iacr.org/2004/008.
3. M. Bellare and P. Rogaway. Entity Authentication and Key Distribution. In *CRYPTO 1993*, volume 773 of *LNCS*, pages 110–125. Springer, 1993.
4. S. Blake-Wilson, D. Johnson, and A. Menezes. Key Agreement Protocols and their Security Analysis. In *IMA Cryptography and Coding 1997*, volume 1335 of *LNCS*, pages 30–45. Springer, 1997.
5. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32(3):585–615, 2003.
6. C. Boyd and K.-K. R. Choo. Security of Two-Party Identity-Based Key Agreement. In *Mycrypt 2005*, volume 3715 of *LNCS*, pages 229–243. Springer, 2005.
7. C. Boyd, W. Mao, and K. Paterson. Deniable Authenticated Key Establishment for Internet Protocols. In *Security Protocols 2003*, volume 3364 of *LNCS*, pages 255–271. Springer, 2005.
8. C. Boyd and D. Park. Public Key Protocols for Wireless Communications. In *ICISC 1998*, volume 1807 of *LNCS*, pages 47 – 57. Springer, 1998. (Available from http://sky.fit.qut.edu.au/∼boydc/papers/icisc98.ps.gz).
9. R. Canetti and H. Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 453–474. Springer, 2001. Extended version available from http://eprint.iacr.org/2001/040.
10. L. Chen and C. Kudla. Identity Based Authenticated Key Agreement Protocols from Pairings. In *CSFW 2003*, pages 219–233. IEEE Computer Society Press, 2003. Corrected version at http://eprint.iacr.org/2002/184.

11. Z. Cheng and L. Chen. On Security Proof of McCullagh-Barreto's Key Agreement Protocol and its Variants. Cryptology ePrint Archive, Report 2005/201, 2005.
12. Z. Cheng, L. Chen, R. Comley, and Q. Tang. Identity-Based Key Agreement with Unilateral Identity Privacy Using Pairings. In *ISPEC 2006*, volume 3903 of *LNCS*, pages 202–213. Springer, 2006.
13. Z. Cheng, M. Nistazakis, R. Comley, and L. Vasiu. On the Indistinguishability-Based Security Model of Key Agreement Protocols-Simple Cases. Cryptology ePrint Archive, Report 2005/129, 2005.
14. Y. J. Choie, E. Jeong, and E. Lee. Efficient Identity-based Authenticated Key Agreement Protocol from Pairings. *Applied Mathematics and Computation*, 162(1):179–188, 2005.
15. K.-K. R. Choo. *Key Establishment: Proofs and Refutations*. Ph.D. Thesis, Queensland University of Technology, 2006. http://adt.library.qut.edu.au/adt-qut/public/adt-QUT20060928.114022/.
16. K.-K. R. Choo. A Proof of Revised Yahalom Protocol in the Bellare and Rogaway (1993) Model. *The Computer Journal*, 50(5):591–601, 2007. Pre-print version available from http://eprint.iacr.org/2007/188.
17. K.-K. R. Choo, C. Boyd, and Y. Hitchcock. Examining Indistinguishability-Based Proof Models for Key Establishment Protocols. In *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 585–604. Springer, 2005.
18. K.-K. R. Choo, C. Boyd, and Y. Hitchcock. On Session Key Construction in Provably Secure Protocols. In *Mycrypt 2005*, volume 3715 of *LNCS*, pages 116–131. Springer, 2005.
19. S. S. M. Chow. Personal Communication with Authors of [18], 29 Apr 2005.
20. S. S. M. Chow, S. M. Yiu, and L. C. K. Hui. Efficient Identity Based Ring Signature. In *ACNS 2005*, volume 3531 of *LNCS*, pages 499–512. Springer, 2005.
21. S. S. M. Chow, S. M. Yiu, L. C. K. Hui, and K. P. Chow. Efficient Forward and Provably Secure ID-Based Signcryption Scheme. In *ICISC 2003*, volume 2971 of *LNCS*, pages 352–369. Springer, 2003.
22. H. Krawczyk. HMQV: A High-Performance Secure Diffie–Hellman Protocol. In *CRYPTO 2005*, volume 3621 of *LNCS*, pages 546–566. Springer, 2005. Extended version available from http://eprint.iacr.org/2005/176.
23. C. Kudla and K. G. Paterson. Modular Security Proofs for Key Agreement Protocols. In *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 549–569. Springer, 2005.
24. S. Kunz-Jacques and D. Pointcheval. About the Security of MTI/C0 and MQV. In Roberto De Prisco and Moti Yung, editors, *Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006, Proceedings*, volume 4116 of *Lecture Notes in Computer Science*, pages 156–172. Springer, 2006.
25. N. McCullagh and P. S. L. M. Barreto. A New Two-Party Identity-Based Authenticated Key Agreement. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 262–274. Springer, 2005. Extended version available from http://eprint.iacr.org/2004/122.
26. David Pointcheval and Jacques Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
27. R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, 2001.
28. R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret: Theory and Applications of Ring Signatures. In *Theoretical Computer Science: Essays in Memory of Shimon Even*, volume 3895 of *LNCS*, pages 164–186. Springer, 2006.
29. D. Samfat, Re. Molva, and N. Asokan. Untraceability in Mobile Networks. In *ACM MobiCom 1995*, pages 26–36. ACM Press, 1995.
30. K.-A. Shim. Efficient ID-based Authenticated Key Agreement Protocol based on Weil Pairing. *IEE Electronics Letters*, 39(8):653–654, 2002.
31. K.-A. Shim. Cryptanalysis of Two ID-based Authenticated Key Agreement Protocols from Pairings. Cryptology ePrint Archive, Report 2005/357, 2005.
32. V. Shoup. On Formal Models for Secure Key Exchange (Version 4). Technical Report RZ 3120 (#93166), IBM Research, Zurich, 1999.
33. N. Smart. An Identity based Authenticated Key Agreement Protocol based on the Weil Pairing. *IEE Electronics Letters*, 38(13):630–632, 2002.
34. H.-M. Sun and B.-T. Hsieh. Security Analysis of Shim's Authenticated Key Agreement Protocols from Pairings. Cryptology ePrint Archive, Report 2003/113, 2003.
35. Y. Wang. Efficient Identity-Based and Authenticated Key Agreement Protocol. Cryptology ePrint Archive, Report 2005/108, 2005.
36. G. Xie. An ID-Based Key Agreement Scheme from Pairing. Cryptology ePrint Archive, Report 2005/093, 2005.
37. X. Yi. Efficient ID-Based Key Agreement from Weil Pairing. *IEE Electronics Letters*, 39(2):206–208, 2003.

A Canetti–Krawczyk Model

We now present a brief overview of the CK model [9]. In the CK model, there are two adversarial models, namely the unauthenticated-links / real world model (UM) and the authenticated-links / ideal world model (AM). AM is the (ideal) world where messages are authenticated magically, and UM is the (real) world in which we want our protocols to be proven secure. Proving protocols secure in the CK model is usually by translating a provably secure protocol in the AM to a provably secure protocol in the UM with the use of an authenticator. However, we will not use this approach as the use of the authenticator can result in a more expensive protocol. Instead the protocol is proven secure in the UM following Krawczyk's approach [22] (in a similar fashion as proof in the BR model).

A.1 Adversarial Power

The adversary, $\mathcal{A}$, controls the communications between the protocol participants by interacting with the set of oracles, $\Pi^i_{U_u,U_v}$, where $\Pi^i_{U_u,U_v}$ is defined to be the i^{th} instantiation of a protocol participant, U_u, in a specific protocol run and U_v is the principal with whom U_u wishes to establish a secret key. $\mathcal{A}$ controls the communication channels via the queries to the targeted oracles. A description of the oracle types is presented as follows.

$\mathsf{Send}(U_u, U_v, i, m)$ **query.** This query to an oracle, $\Pi^i_{U_u,U_v}$, computes a response according to the protocol specification and decision on whether to accept or reject yet, and returns them to the adversary $\mathcal{A}$. If $\Pi^i_{U_u,U_v}$ has either accepted with some session key or terminated, this will be made known to $\mathcal{A}$. Note that if $m = *$, then this will result in the instantiation of the oracle $\Pi^i_{U_u,U_v}$ if such an oracle has not been created previously.

$\mathsf{Session\text{-}Key\ Reveal}(U_u, U_v, i)$ **query.** Any oracle, $\Pi^i_{U_u,U_v}$, upon receiving such a query and if $\Pi^i_{U_u,U_v}$ has accepted and holds some session key, will send this session key back to $\mathcal{A}$. As Blake-Wilson, Johnson and Menezes [4] have indicated, the Reveal query is designed to capture this notion. Note that this query is known as a $\mathsf{Reveal}(U_u, U_v, i)$ query in the BR model [3].

$\mathsf{Session\text{-}State\ Reveal}(U_u, U_v, i)$ **query.** The oracle, $\Pi^i_{U_u,U_v}$, upon receiving such a query and if $\Pi^i_{U_u,U_v}$ has neither accepted nor held some session key, will return all its internal state to $\mathcal{A}$. This includes any ephemeral parameters but not long-term secret parameters.

$\mathsf{Corrupt}(U_u)$ **query.** This query captures unknown key share attacks and insider attacks. This query allows $\mathcal{A}$ to corrupt the principal U_u at will, and thereby learn the complete internal state of the corrupted principal. Notice that a $\mathsf{Corrupt}$ query does not result in the release of the session keys since $\mathcal{A}$ already has the ability to obtain session keys through Reveal queries.

Protocols proven secure in the model that allows the $\mathsf{Corrupt}$ query are also proven secure against the unknown-key share attack. That is, if a key is to be shared between some parties, U_1 and U_2, the corruption of some other (non-related) player in the protocol, say U_3, should not expose the session key shared between U_1 and U_2 [17].

$\mathsf{Test}(U_u, U_v, i)$ **query.** This query is the only oracle query that does not correspond to any of $\mathcal{A}$'s abilities or any real-world event. If $\Pi^i_{U_u,U_v}$ has accepted with some session key and is being asked a $\mathsf{Test}(U_u, U_v, i)$ 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. Informally, $\mathcal{A}$ succeeds if $\mathcal{A}$ can guess the bit b.

A.2 Partnership

Partnership in the CK model is defined using the notions of matching sessions and session identifiers (SIDs), as described in Definition 4. There is no formal definition of how SIDs should be defined as highlighted in [16]. It is assumed that SIDs are known by protocol participants before the protocol begins. Such an assumption might not be practical, as it requires some form of communication between the protocol participants prior to the start of the protocol. In practice, SIDs may be determined during protocol execution.

Definition 4 (Matching Sessions [9]). *Two sessions are said to be matching if they have the same session identifiers and corresponding partner identifiers.*

A.3 Freshness

Freshness is defined in Definition 5.

Definition 5. *An oracle, $\Pi^i_{U_1}$, is* fresh *at the end of its execution if:*

- *$\Pi^i_{U_1}$ and its partner $\Pi^j_{U_2}$ (if such a partner exists) have not been asked any* Reveal *queries, and*
- *both principals U_1 and U_2 have not been asked any* Corrupt *queries.*

A.4 Security Model

Security in the model is defined using the game $\mathcal{G}$, played between a malicious adversary $\mathcal{A}$ and a collection of $\Pi^i_{U_u,U_v}$ oracles for players U_u, U_v and instances i. $\mathcal{A}$ runs the game $\mathcal{G}$, with the following settings.

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. Note that the test session chosen must be fresh. Depending on a randomly chosen bit b, $\mathcal{A}$ is given either the actual session key or a session key drawn randomly from the session key distribution.
Stage 3: $\mathcal{A}$ continues making any oracle queries at will but cannot make Corrupt and/or Reveal that trivially expose the test session key.
Stage 4: Eventually, $\mathcal{A}$ terminates the game simulation and outputs a bit b', which is its guess of the value of b.

The 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 $\mathsf{Test}(U_u, U_v, i)$ query, where $\Pi^i_{U_u,U_v}$ is fresh and has accepted, $\mathcal{A}$'s guess bit b' equals the bit b selected during the $\mathsf{Test}(U_u, U_v, i)$ query.

Let the advantage function of $\mathcal{A}$ be denoted by $\mathsf{Adv}^{\mathcal{A}}(k)$, where $\mathsf{Adv}^{\mathcal{A}}(k) = |2 \times Pr[b = b']| - 1$. We now define security in the model as described in Definition 6.

Definition 6 (Security). *A protocol is secure in the CK model if,*

Validity. *If two uncorrupted oracles complete matching sessions, then both oracles must hold the same session key.*
Indistinguishability. *For all probabilistic, polynomial time adversaries, $\mathcal{A}$, the advantage of $\mathcal{A}$, $\mathsf{Adv}^{\mathcal{A}}(k)$, in game $\mathcal{G}$ is negligible.*

In the CK model, a party can no longer interact with any other protocol participants once it has been corrupted. Therefore, one cannot make any security guarantees about future sessions associated with a corrupted party and it follows easily that key compromise impersonation (KCI) resistance is not considered. Below recalls the formal definition for KCI resistance given by Krawczyk.

Definition 7 (Security Against KCI [22]). *We say that an adversary, $\mathcal{A}$, with access to the private key of party A (but not the private key of party B) succeeds in a KCI-resistance attack against A, if $\mathcal{A}$ is able to distinguish from random the session key of a complete session at A for which the session peer is uncorrupted and the session and its matching session (if it exists) are clean (i.e., $\mathcal{A}$ does not have access to the session's state at the time of session establishment nor has $\mathcal{A}$ issued any* Reveal *query against the session upon its completion).*

B Security Proofs for Proposed Signature Schemes

Borrowing the idea of the forgery game from Krawczyk [22], we define below the existential unforgeability against adaptive-chosen-message-and-identity-attack of our pairing challenge-response signature scheme.

To simplify the discussion, we first describe a simplified variant of pairing challenge response signature without validity checking of the challenge and the user of private key for verification; and we prove its security by building a BDH solver $\mathcal{S}$ from a forger $\mathcal{F}$.

The unforgeability of the strong scheme in Section 4.2 can be proven in a similar manner by relying on another intractability assumption. Finally, we discuss how to extend the security notion of the basic scheme to that of the dual signature and show its security. Savvy readers will find the below proof and the proof for our key agreement protocols share many similarities.

B.1 Unforgeability of Identity-based Challenge-Response Signatures

Definition 8 (EUF-PCR-CMIA2). *For an ID-based challenge-response signature scheme, the existential unforgeability against adaptive-chosen-message-and-identity-attack is defined in the following game between a challenger $\mathcal{S}$ and an adversary $\mathcal{F}$:*

1. *System parameters generated according to the* Setup *procedure (excluding the* master secret*) and the challenge W^* is given to $\mathcal{F}$.*
2. *$\mathcal{F}$ can adaptively choose a polynomial number of oracle queries and ask $\mathcal{S}$ for answers.*
   (a) Extraction oracle $\mathcal{XO}$: Given an identity ID, the corresponding private key S_{ID} is returned.
   (b) Signing oracle $\mathcal{SO}$: Given a challenge-identity-message tuple (W, ID, m), the corresponding signature σ is returned.
3. *$\mathcal{F}$ either halts with* fail*, or outputs a signature-identity-message tuple $(\sigma^*, \mathsf{ID}^*, m^*)$ (for verifier ID').*
4. *$\mathcal{F}$ wins if all of the following conditions hold.*
   (a) The signature σ is a valid challenge-response signature of ID^* on message m^* with respect to challenge W^* (for verifier ID').
   *(b) The identity ID^* (*not *ID') did not appear in any extraction oracle $\mathcal{XO}$ query.*
   (c) The pair $(W^*, \mathsf{ID}^*, m^*)$ did not appear in any signing oracle $\mathcal{SO}$ query (for verifier ID').

B.2 Basic Scheme

Setup, Extract : Same as the strong scheme described in Section 4.

Sign: We suppose B is the verifier requesting for a signature on message m from the signer A.

1. B picks $b \in_R \mathbb{Z}_q^*$ and sends $W_B = bP$ to A.
2. A picks $a \in_R \mathbb{Z}_q^*$ and sends $W_A = aQ_A$ and $e_A = \hat{e}((a + h_A)S_A, W_B)$ to B where $h_A = \mathcal{H}_0(W_A, m)$.

Verify: As the strong scheme, bP_{pub} is used instead of bS_B.

B.3 Proof for Unforgeability

Given a forger $\mathcal{F}$ of our basic scheme, below show how to build a BDH solver $\mathcal{S}$.

Setup : Suppose the problem instance is (P, xP, yP, zP), the public key of the KGC is set as xP and the challenge W^* is set as zP.

$\mathcal{H}$ queries : If an $\mathcal{H}$ query is previously asked, then the stored answer in the list $L_{\mathcal{H}}$ will be returned. Suppose the J^{th} distinct $\mathcal{H}$ query is ID_J, then $\mathcal{S}$ responses with yP; otherwise, $\mathcal{S}$ chooses $r_i \in_R \mathbb{Z}_q^*$, stores it in the list $L_{\mathcal{H}}$ along with ID_I, and outputs r_iP.

$\mathcal{H}_0$ *queries* : $\mathcal{S}$ maintains a list $L_{\mathcal{H}_0}$ to ensure that previously asked queries will receive the same answer. However, special value may be plugged into the list in the simulation of the $\mathcal{SO}$ queries with ID_I as the signer.

$\mathcal{XO}$ *queries* : The simulation fails (event 0) if the request is ID_J, otherwise the corresponding r_i is retrieved from the list $L_{\mathcal{H}}$ and $r_i(xP)$ is returned.

$\mathcal{SO}$ *queries* : Suppose W_V is received as the challenge and the designated verifier is ID_V. For any signing query of ID_I, $\mathcal{S}$ knows the corresponding private key, so the simulation can be done as a typical protocol invocation. Except for the following special handling for ID_J, α and h are chosen randomly from $\mathbb{Z}_q^*$ and setting $h = \mathcal{H}_0(\alpha P - hQ_J, m)$. If $\mathcal{H}_0(\alpha P - hQ_J, m)$ is previously queried, another α is chosen. The signature (W_J, e_J) can be computed by $W_J = \alpha P - hQ_J$ and $e_J = \hat{e}(\alpha xP, W_V)$. It is easy to see the signature is valid since $\hat{e}(x(W_J + hQ_J), W_V) = \hat{e}(\alpha xP, W_V)$.

Forgery : Suppose $\mathcal{F}$ does not halt, now $\mathcal{S}$ returns $\sigma^* = (W_J^*, e_J^*)$ as a valid signature. If it is not made on behalf of ID_J, the simulation fails. Event 0 would not occur if ID_J were chosen by $\mathcal{F}$ as the target of attack, such choice is made with probability $1/N_{\mathcal{Q}}$ where $N_{\mathcal{Q}}$ is the number of $\mathcal{H}$ queries.

Suppose the simulation does not abort we have $e_J^* = \hat{e}(W_J^* + h_J^* yP, zP)^x$ where $h_J^* = \mathcal{H}_0(W_J^*, m^*)$. We ignore the small probability that $\mathcal{F}$ can correctly guess the value of $\mathcal{H}_0(W_J^*, m^*)$ without making the corresponding $\mathcal{H}_0$ query. Now $\mathcal{S}$ runs $\mathcal{F}$ for a second time with the same settings except setting the response of $\mathcal{H}_0$ query of (W_J, m^*) as h'_J. By the standard forking lemma argument [26], in this second time $\mathcal{F}$ gives a valid forgery with $e'_J = \hat{e}(W_J^* + h'_J yP, zP)^x$. The solution of the BDH problem is given by $(e'_J/e_J^*)^{(h'_J - h_J^*)^{-1}} = \hat{e}(yP, zP)^x$.

B.4 Strong Scheme

For our ID-based strong challenge-response signature, the simulator $\mathcal{S}$ needs to provide W'^* in addition to W^* which satisfy the relation $\hat{e}(W^*, W'^*) = \hat{e}(Q_B, Q_A)$ for certain users A and B. This requires the forgery game of our basic scheme to be modified such that the adversary can choose two identities A and B after simulator $\mathcal{S}$ outputted the system parameter. The security of our strong scheme depends on a variant of BDH problem of (P, xP, yP, zP), in which $yz^{-1}P$ is also included in the problem instance. It is easy to see that the proof for our basic scheme works equally well since a valid W'^* can be returned by returning $yz^{-1}P$, if $\mathcal{S}$ sets $\mathcal{H}(\mathsf{ID}_A) = yP$.

B.5 Dual Scheme

Signatures produced by the dual scheme is in the following form

$$(W_A = aQ_A, W_B = bQ_B, e = \hat{e}(Q_A, Q_B)^{s(a+h_A)(b+h_B)})$$

which can be seen as two signatures (W_A, e) and (W_B, e) of the underlying (single) scheme. If $h_A = \mathcal{H}_0(W_A, m_A)$ and $h_B = \mathcal{H}_0(W_B, m_B)$, the former signature is one produced by A on message m_A under the challenge $W_B + h_B Q_B$ while the later is one produced by B on message m_B under the challenge $W_A + h_A Q_A$. Our security notion for ID-based scheme can be naturally extended to our ID-based dual challenge-response signature's case. More precisely, in the forgery game, the signing oracle query should include one more parameter that is the message to be signed by the peer, and a successful forgery comes with an extra message that is chosen by the forger $\mathcal{F}$.

The simulation for our basic scheme still works in this modified forgery game, but the solution of the BDH problem should be computed in another way since the forgery comes with an extra component. With the notation in the proof for basic scheme and suppose the dual signature given by the forger $\mathcal{F}$ is the one by J and V on message m_J and m_V respectively; $e_J^* = \hat{e}(W_J^* + h_J^* yP, zP + h_V^* Q_V)^x$ where $h_J^* = \mathcal{H}_0(W_J^*, m_J^*)$ and $h_V^* = \mathcal{H}_0(W_V^*, m_V^*)$. The second run gives $e'_J = \hat{e}(W_J^* + h'_J yP, zP + h'_V Q_V)^x$ where $h'_J = \mathcal{H}_0(W_J^*, m_J^*)$ and

$h'_V = \mathcal{H}_0(W^*_V, m'_V)$. Note that forger $\mathcal{F}$ may give another message m'_V in the second run since it is something which is under $\mathcal{F}$'s control. e'_J/e^*_J gives $\hat{e}((h'_J - h^*_J)yP, zP)^x \cdot \hat{e}((h'_J - h^*_J)yP, h'_V Q_V)^x / \hat{e}((h^*_J - h^*_J)yP, h^*_V Q_V)^x$. Notice that Q_V is in the form of $r_V P$, so xQ_V can be computed as $r_V(xP)$ by simulator $\mathcal{S}$. Thus the last two term of the above expression can be cancelled out which leaves the term $\hat{e}((h'_J - h^*_J)yP, zP)^x$. BDH problem can be solved in a similar way as the proof for our basic scheme.

C Security Proofs for Escrow-Free Protocols

Proof (Theorem 3). For brevity we only highlight the changes that should be made on the indistinguishability proof for our basic protocol. Recall that in additional to $W_A = aQ_A$ ($W_B = bQ_B$), an additional element $T_A = aP$ ($T_B = bP$) is included in the message flow.

Send *queries (*ID_I *as initiator and* ID_J *as responder)* : For the N^{th} invocation, $W_{I,N} = r_I\tau(zP)$ is responded, since $Q_I = r_I P$, the implicit ephemeral secret is τz. The corresponding $T_{I,N}$ can be computed by $\tau(zP)$.

Send *queries (*ID_J *as initiator and* ID_K *as responder)* : Now $W_{J,\ell} = \alpha_\ell P - h_{J,\ell} Q_J$, the implicit ephemeral secret is $\alpha_\ell y^{-1} - h_{J,\ell}$, the corresponding $T_{J,\ell}$ can be computed by $\alpha_\ell(y^{-1}P) - h_{J,\ell}P$.

These two are the only necessary changes for $\mathcal{S}$'s simulation. Of course what $\mathcal{S}$ can solve is MBDH problem but not BDH problem, since $y^{-1}P$ is needed. □

Proof (Theorem 4). The proof for weak KGC-forward-secrecy is similar to that of Chen and Kudla [10]. We construct a simulator $\mathcal{S}$ that solve the CDH problem (P, xP, yP) with the help of an adversary $\mathcal{A}$ which has a non-negligible advantage against the security of our protocol described in Figure 3.

Setup : A random s is chosen from $\mathbb{Z}^*_q$ and given to $\mathcal{A}$, KGC's public key is sP.

$\mathcal{H}$ *queries* : $\mathcal{S}$ chooses $r_i \in_R \mathbb{Z}^*_q$, stores (r_i, ID_i) in the list $L_{\mathcal{H}}$, and outputs $r_i P$.

$\mathcal{H}_0$ *queries* : $\mathcal{S}$ maintains a list $L_{\mathcal{H}_0}$ to ensure the random oracle property.

$\mathcal{K}$ *queries* : Maintains the random oracle properties, with the help of a list $L_{\mathcal{K}}$.

Corrupt *queries* : With s, $\mathcal{A}$ can compute easily by itself.

Send *queries* : For $\Pi^k_{I,J}$, $\mathcal{S}$ answers $W_I = t_k a Q_I = t_k r_I(aP)$ and $T_I = t_k(aP)$. For $\Pi^\ell_{J,I}$, $\mathcal{S}$ answers $W_J = t_\ell b Q_J = t_\ell r_J(bP)$ and $T_J = t_\ell(bP)$.

Key Reveal *and* State Reveal *queries* : Fully supported, except $\Pi^k_{I,J}$ and $\Pi^\ell_{J,I}$.

Test *queries* : $\mathcal{S}$ aborts if $\mathcal{A}$ does not choose the oracle $\Pi^k_{I,J}$. Otherwise, $\Pi^k_{I,J}$ must have accepted after having had a matching conversation with another oracle, if it happens to be $\Pi^\ell_{J,I}$, the second component of the keying material is $t_k t_\ell(abP)$. Note that $\mathcal{S}$ cannot compute this value by itself (without $\mathcal{A}$ helping), so it cannot returns a real session key, and thus the only choice is to return a random key drawn from session key distribution (range of $\mathcal{K}$).

Solving the CDH problem : If $\mathcal{S}$ does not abort and $\mathcal{A}$ is so clever (with probability $\epsilon(k)$) that can distinguish between real session key and random session key, $\mathcal{A}$ must have queried the key derivation oracle $\mathcal{K}$ for the keying material $(*, t_k t_\ell(abP))$ (where $*$ is something does not really matter) to output its guess correctly. Now $\mathcal{S}$ randomly chooses one of $\mathcal{A}$'s $\mathcal{K}$'s queries $(*, \pi)$. If $\mathcal{S}$ is lucky enough that π is the above keying material, $\mathcal{S}$ outputs the solution as $\pi/(t_k t_\ell)$.

Probability analysis : Considering the events above, $\mathcal{S}$ wins the game with probability $\epsilon(k)/(N_{\mathcal{C}}^2 N_{\mathcal{K}})$, where $N_{\mathcal{C}}$ is the number of sessions created and $N_{\mathcal{K}}$ is the number of key derivation oracle queries. □

Proof (Theorem 5). As the KCIR proof for our basic protocol, $\mathcal{S}$ can correctly answer Corrupt query of ID_I, the initiating party. Hence KCIR follows. □

D Security Proofs for Anonymous Protocols

Proof (Theorem 6). We construct a simulator $\mathcal{S}$ that wins the interactive game with a BDH challenger (the BDH problem instance is (P, xP, yP, zP) and the last part of the challenge is h) with the help of an adversary $\mathcal{A}$ that can break our protocol.

Setup : KGC's public key is xP.

$\mathcal{H}$ *queries* : $\mathcal{S}$ embeds yP in the answer of many $\mathcal{H}$ queries, which depends on the result of flipping a coin $W \in \{0,1\}$ yielding 0 with probability ζ (to be determined) and 1 with probability $1-\zeta$.

Denote the i-th distinct $\mathcal{H}$ query by ID_i. $\mathcal{S}$ chooses $r_i \in_R \mathbb{Z}_q^*$. If $W = 0$, r_iP is returned; otherwise, $\mathcal{S}$ responses with $r_i(yP)$. For both cases r_i and W are stored it in the list $L_{\mathcal{H}}$ along with ID_i.

$\mathcal{H}_0$ *queries* : $\mathcal{S}$ maintains a list $L_{\mathcal{H}_0}$ to ensure the random oracle property. However, special value may be plugged into the list in the simulation of Send queries when $\mathcal{S}$ cannot compute the private key of all of the members of a ring.

$\mathcal{K}$ *queries* : Maintains the random oracle properties, with the help of a list $L_{\mathcal{K}}$.

Corrupt *queries* : The simulation fails if the coin value stored in the list $L_{\mathcal{H}}$ along with ID_i is 1; otherwise corresponding r_i is retrieved and $r_i(xP)$ is returned.

Send *queries* : Suppose the initiating ring A is of size n_1, the responding ring B is of size n_2, ψ_A and ψ_B are the unique value used by the initiator and the responder respectively. For the session creation, we consider the following cases.

1. At least one member of the responding ring has the public key in the form of r_iP, which further divided into two cases.
   (a) At least one member of the initiating ring has the public key in the form of r_iP: In this case, $\mathcal{S}$ can compute at least one private key among those in the initiating ring, $\mathcal{S}$ can simply simulate the protocol as normal.
   (b) All members of the initiating ring have the public key in the form of $r_i(yP)$: $\mathcal{S}$ manipulates the random oracle $\mathcal{H}_0$ as follows.
      i. Chooses an index $s \in_R \{1, 2, \cdots, n_1\}$.
      ii. Chooses $U_i \in_R \mathbb{G}$, computes $h_i = \mathcal{H}_0(U_i, B, \psi_A)$ $\forall i \in \{1, 2, \cdots, n_1\}\backslash\{s\}$.
      iii. Chooses $h'_s \in_R \mathbb{Z}_q^*$ and $t \in_R \mathbb{Z}_q^*$, computes $U_s = tP - h'_s Q_{ID_s} - \sum_{i\neq s}\{U_i + h_i Q_{ID_i}\}$.
      iv. Stores the relationship $h'_s = \mathcal{H}_0(U_s, B, \psi_A)$ to the list $L_{\mathcal{H}_0}$ and stores $T = t(xP)$ into an auxiliary list as an auxiliary data corresponding to this session. If collision occurs (which is not likely since with high probability ψ_A does not repeat), repeats step iii.
2. All members in the responding ring have the public key in the form of $r_i(yP)$, which further divided into three cases.
   (a) No member in the responding ring has a public key in the form of $r_i(yP)$: For first such query, $\mathcal{S}$ embeds the hard problem as follows, otherwise it proceeds as case 1a.
      i. Chooses an index $s \in_R \{1, 2, \cdots, n_1\}$, retrieves (r_s, ID_{A_s}) from $L_{\mathcal{H}}$.
      ii. $\forall i \in \{1, 2, \cdots, n_1\}\backslash\{s\}$, takes $u_i \in_R \mathbb{Z}_q^*$ and keeps a record; then computes $U_i = u_iP \in \mathbb{G}$.
      iii. Computes $h_i = \mathcal{H}_0(U_i, B, \psi_A)$ $\forall i \in \{1, 2, \cdots, n_1\}\backslash\{s\}$.

iv. Chooses $\tau \in_R \mathbb{Z}_q^*$, and computes $U_i = r_s\tau(zP)$, τ is recorded.

v. $\mathcal{A}$ will keep on asking the value of $c_i = \mathcal{H}_0(V_i, A, \psi_B)$, $\mathcal{S}$ answers as normal except for the last query $\mathcal{H}_0(V_{n_2}, A, \psi_B)$.

vi. When $\mathcal{S}$ has collected all V_is for a single session (which is linked by ψ_B), we give $X = \sum_{i\in\{1,2,\cdots,n_2\}} V_i + \sum_{i\in\{1,2,\cdots,n_2-1\}} c_iQ_{B_i}$ to the interactive BDH challenger. If (V_{n_2}, A, ψ_B) can be found in list $L_{\mathcal{H}_0}$ (which is unlikely as argued before), the simulation fails.

vii. $\mathcal{S}$ dumps all maintained lists and system parameters to the tape σ and outputs (X, σ). The interactive BDH challenger returns $h \in_R \mathbb{Z}_q^*$.

viii. $\mathcal{S}$ reconstructs all the lists and system parameters from σ, retrieves $(r_{n_2}, ID_{B_{n_2}})$ from $L_{\mathcal{H}}$ and set $\mathcal{H}_0(V_{n_2}, A, \psi_B) = h{r_{n_2}}^{-1}$.

(b) At least one in the responding ring holds a public key in the form of r_iP (excluding the above case that all members hold keys in r_iP form): $\mathcal{S}$ performs exactly the same simulation as in case 1a.

(c) All members of the initiating ring have the public key in the form of $r_i(yP)$: $\mathcal{S}$ performs exactly the same simulation as in case 1b.

If the adversary acts as the initiating party, the simulation can also be done in a similar manner as above, with the criterion on the initiator ring and the responding ring interchanged. However, $\mathcal{S}$ will not embed the hard problem in the simulation for case 2a, instead the simulation is done in the way as in case 2b (which is the same as case 1a).

Key Reveal *queries* :

1. For case 1a and 2b, it is trivial to compute the session key.
2. For case 1b and 2c, it is easy to see that $\mathcal{S}$ can use T from the corresponding entry in the auxiliary list to compute the session key.

State Reveal *queries* :

1. For case 1a and 2b, it is trivial to reveal the session state.
2. For case 1b and 2c, $\mathcal{S}$ fails, which is our limitation in revealing session state.

By the game's rule, the adversary will not make Key Reveal or State Reveal queries for case 2a.

Test *queries* : If the session prepared in case 2a is selected, the session key is in the following form.

$$\begin{aligned}
&\mathcal{K}(\hat{e}(\sum_{i=1}^{n_1}(U_i + h_jQ_{A_j}), \sum_{i=1}^{n_2}(V_i + c_iQ_{B_i}))^x) \\
&= \mathcal{K}(\hat{e}(\sum_{i\in\{1,2,\cdots,n_1\}\setminus\{s\}}(U_i + h_jQ_{A_j}) + h_sQ_{ID_s} + r_s\tau(zP), \\
&\qquad \sum_{i\in\{1,2,\cdots,n_2-1\}}(V_i + c_iQ_{B_i}) + V_{n_2} + hyP)^x) \\
&= \mathcal{K}(\hat{e}(wP + r_s\tau(zP), X + hyP)^x) \\
&\quad (w \text{ can be computed with the help of } u_i\text{s and} r_i\text{s previously recorded}) \\
&= \mathcal{K}(\hat{e}(wP, X + hyP)^x\hat{e}(zP, X + hyP)^{xr_s\tau}) \\
&= \mathcal{K}(\hat{e}(w(xP), X + hyP)\hat{e}(xP, X + hyP)^{zr_s\tau}).
\end{aligned}$$

Note that $\mathcal{S}$ cannot compute $\hat{e}(xP, X + hyP)^{zr_s\tau}$ by itself, so it cannot returns a real session key, thus the only choice is to return a random key drawn from session key distribution (range of $\mathcal{K}$).

Answering interactive BDH challenger : If $\mathcal{S}$ does not abort and $\mathcal{A}$ is so clever (with probability $\epsilon(k)$) that can distinguish between real session key and random session key, $\mathcal{A}$ must have queried the key derivation oracle $\mathcal{K}$ for the keying material $\hat{e}(w(xP), X + hyP)\hat{e}(xP, X + hyP)^{zr_s\tau}$ to output its guess correctly. Now $\mathcal{S}$ randomly chooses one of $\mathcal{A}$'s $\mathcal{K}$'s queries π. If $\mathcal{S}$ is lucky enough (with probability $1/N_{\mathcal{K}}$, where $N_{\mathcal{K}}$ is the number of key derivation oracle queries) that π is the above keying material, $\mathcal{S}$ answers the interactive BDH challenger correctly with $(\pi/\hat{e}(w(xP), X + hyP))^{1/(r_s\tau)}$.

Probability analysis : For Corrupt, the simulation would not fail if all $N_{\mathcal{E}}$ such queries correspond to coin value $W = 0$, the probability of such event is $\zeta^{N_{\mathcal{E}}}$.

For $\mathcal{S}$ to embed the hard problem successfully, we need case 2a to happen for at least one session. Let the initiating ring is of size n_1 and the responding ring B is of size n_2. Case 2a happens with probability $\zeta^{n_1}(1-\zeta^{n_2})$, ignoring the fact that each identity should be independent – which only results in a small error in probability if n_1 and n_2 are large. For brevity we assume $n_1 = n_2 = n$ and let $\gamma = \zeta^n(1 - \zeta^n)$. Suppose $N_{\mathcal{C}}$ is the number of sessions created. Assuming the choice of the initiating ring and the responding ring for each of these sessions are independent, the probability that $\mathcal{S}$ can successfully embed the hard problem is $1 - (1 - \gamma)^{N_{\mathcal{C}}}$, which is lower bounded by $N_{\mathcal{C}} \cdot \gamma$.

For $\mathcal{S}$ to solve the hard problem successfully, $\mathcal{A}$ must select the session that $\mathcal{S}$ embedded the hard problem as the test session, which happens with probability $1/N_{\mathcal{C}}$. Combing all probability, a successful simulation occurs with probability $f(\zeta)\epsilon(k)/N_{\mathcal{K}}$, where $f(\zeta) = (\zeta^{N_{\mathcal{E}}})(N_{\mathcal{C}}\zeta^n(1 - \zeta^n))/N_{\mathcal{C}} = \zeta^{N_{\mathcal{E}}+n}(1 - \zeta^n)$. A simple differentiation shows that $f(\zeta)$ is maximized at $\zeta = (\frac{N_{\mathcal{E}}+n}{N_{\mathcal{E}}+2n})^{1/n}$, this value of ζ maximizes the probability of $\mathcal{S}$ to win the interactive BDH challenger. □

Proof (Theorem 7). Suppose s indexes the real protocol participant among the identities set $\{A_1, A_2, \cdots, A_n\}$. All U_is for $i \in \{1, 2, \cdots, n\}\backslash\{s\}$ are randomly chosen, so does U_s since r'_s is randomly chosen from $\mathbb{Z}_q^*$. Anyway these elements leave no track of real identity. Element related to the real identity is only involved after the key is established. However, $(r'_S + h_S)S_{A_s}$, together with those U_is, constitutes a Chow *et al.*'s ring signature [20], which is proven to be unconditionally signer-anonymous, meaning participant A_s is hidden in our setting. □

E Previously Unpublished Attack

Figure 4 describes Yi's protocol [37], $x(W)$ means the x-coordinate of point W.

$$
\begin{array}{ccc}
\text{Initiator } A & & \text{Responder } B \\
(Q_A := \mathcal{H}(\mathsf{ID}_A), S_A := sQ_A) & & (Q_B := \mathcal{H}(\mathsf{ID}_B), S_B := sQ_B) \\
 & & \\
a \in_R \mathbb{Z}_q^*; W_A := aQ_A & & \\
 & \xrightarrow{ID_A, W_A} & b \in_R \mathbb{Z}_q^*; W_B := bQ_B \\
h_A := x(W_A) & \xleftarrow{ID_B, W_B} & h_A := x(W_B) \\
\multicolumn{3}{c}{sk_A = \hat{e}((a + x(W_A))Q_A, (b + x(W_B))Q_B)^s = sk_B}
\end{array}
$$

Fig. 4. Yi's identity-based key agreement protocol

Adversary $\mathcal{A}$ learns the session key of a fresh session of Yi's protocol as follows.

1. $\mathcal{A}$ asks a Corrupt query to C prior to the execution of the protocol – static corruption. $\mathcal{A}$ now runs as C.
2. A sends $W_A = aQ_A$ to B, which is intercepted by C.
3. C picks $r \in_R \mathbb{Z}_q^*$, computes $W_C = W_A + x(W_A)Q_A + rP$, and sends W_C to B impersonating A.

4. B responds with W_B as per protocol specification.
5. B computes the session key, $sk_B = \hat{e}(W_C + x(W_C)Q_C, (b + x(W_B))S_B)$.
6. Note that B and A are non-partners as the message received by B, W_C, is not being sent by A. Hence, the adversary is able to reveal B's session key with a Reveal query.
7. C forwards W_B to A impersonating B.
8. A computes the session key as $sk_A = \hat{e}((a + x(W_A))S_A, W_B + x(W_B)Q_B)$.
9. The adversary now computes the following.

$$\begin{aligned}
&\frac{sk_B}{\hat{e}(rP_{pub} + x(W_C)S_C, W_B + x(W_B)Q_B)}\\
&= \frac{\hat{e}(W_C + x(W_C)Q_C, (b + x(W_B))S_B)}{\hat{e}(rP_{pub} + x(W_C)S_C, W_B + x(W_B)Q_B)}\\
&= \frac{\hat{e}(W_A + x(W_A)Q_A + rP + x(W_C)Q_C, (b + x(W_B))S_B)}{\hat{e}(rP + x(W_C)Q_C, W_B + x(W_B)S_B)}\\
&= \hat{e}(W_A + x(W_A)Q_A + rP + x(W_C)Q_C - rP - x(W_C)Q_C, (b + x(W_B))S_B)\\
&= \hat{e}(W_A + x(W_A)Q_A, (b + x(W_B))S_B)\\
&= \hat{e}((a + x(W_A))Q_A, (b + x(W_B))S_B)\\
&= \hat{e}((a + x(W_A))S_A, (b + x(W_B))Q_B)\\
&= \hat{e}((a + x(W_A))S_A, W_B + x(W_B)Q_B)\\
&= sk_A
\end{aligned}$$