Identity-Based Identification Schemes from ID-KEMs

Prateek Barapatre and Chandrasekaran Pandu Rangan

Theoretical Computer Science Lab.,
Department of Computer Science and Engineering,
IIT Madras, Chennai, India
{pbarapatre.64,prangan55}@gmail.com

Abstract. Identity-based identification(IBI) schemes are means to achieve entity identification in the identity-based setting in a secure fashion. Quite a large number of IBI schemes exist, but, there is still a need for more efficient(in terms of computation and communication) IBI schemes, especially in domains like mobile devices and smart cards. We propose a generic framework for constructing an IBI scheme from an Identity-Based Key Encapsulation Mechanism(ID-KEM) which is semantically secure against adaptive chosen ciphertext attack on one-wayness(OW-CCA2). The derived IBI scheme will be secure against impersonation under active and concurrent attacks. This framework if applied to ID-KEM can lead to more efficient IBI scheme, as opposed to an IBI scheme developed from scratch, depending on the underlying ID-KEM used. Additionally, we propose a new concrete and efficient IBI scheme secure against concurrent attack based on the q-BDHI hard problem assumption.

Keywords: Identity-based cryptography, Identity-based identification, Key encapsulation mechanism, q-Bilinear Diffie-Hellman inversion, Random oracle model.

1 Introduction

A *Standard Identification(SI)* scheme is an interactive protocol which allows an honest entity, called a *prover*, to prove its identity to a verifying entity, called a *verifier*, in a secure fashion. In this process the secret information of the proving entity is not revealed but the *verifier* gets convinced of the genuineness of the *prover*. It enables a *prover* to convince a *verifier* that he is indeed the same entity which he claims to be. The identification schemes can be both symmetric and asymmetric. We concentrate on asymmetric or the public identity-based identification scheme in this work.

Historically there are basically two types of identification schemes:

1. Challenge-and-Response type, obtained in a natural way from encryption or signature schemes.
2. Σ-protocol type, which is a kind of proof of knowledge [1] consisting of a three-round interactive protocol.

B. Gierlichs, S. Guilley, and D. Mukhopadhyay (Eds.): SPACE 2013, LNCS 8204, pp. 111–129, 2013.

In this work, we focus on Challenge-and-Response type of identification scheme using key encapsulation mechanism. The notion of SI can be extended to the ID-based cryptography setting as proposed by Shamir [2] and is known as identity-based identification(IBI) scheme. In an IBI scheme, there is a central authority called the Private Key Generator(PKG), which generates master secret key, master public key and also generates private key for users from their respective public identity. The *prover*, then can verify himself to a *verifier* using a protocol, where the *verifier* knows only the claimed public identity of the *prover*, the public parameters and the master public key. Identification schemes have a large number of applications in the real world, like in the verification of smart cards, military and defense activities, etc.

To break an identification protocol, the aim of an adversary(impersonator) is to prove himself as an entity which he is not. For this the adversary stands between a *verifier* and a *prover*, and invokes many instances of the *prover* application(*prover* clones), each clone having independent states and random tapes. Interacting in some cheating way, the adversary tries to collect information of the secret key from the *prover* clones while the adversary interacts with the *verifier* simultaneously trying to impersonate as the *prover*. This type of an adversary is called a concurrent active attacker.

Recently, Ananda and Arita [3] proposed a framework to obtain identification schemes from key encapsulation mechanism. We extend their idea to the identity-based settings, where the public key of each party is the entity's identity. Moreover, we also show that one-way CCA2 secure ID-KEM serves the required purpose to achieve security against concurrent attacks for IBI schemes. The generic framework proposed can be applied to any OW-CCA2 secure ID-KEM to obtain an IBI scheme secure against concurrent attacks. The KEM-based identification scheme is advantageous as compared to encryption-based identification scheme. This is because, in the case of KEM, an entity can encapsulate random strings which can be generated by the entity itself, as opposed to the encryption-based scheme where one has to encrypt strings(messages) given as input from the other entity. Consequently, KEM-based schemes have a possibility of having simpler and more efficient protocol than encryption-based identification scheme. There are many IBI schemes existing in the literature like those in [4,5,6,7]. With the increased use of mobile devices there is a need for more efficient and faster cryptographic schemes. In the later half of the paper, we propose a new concrete IBI scheme, and our new scheme is more efficient than the existing schemes(in terms of computation or communication).

Our Contribution. In this paper, we first present our general framework for constructing an IBI scheme secure against concurrent attacks from an OW-CCA2 secure ID-KEM. A *verifier* of the OW-CCA2 secure ID-KEM makes a pair of random strings and its corresponding cipher text using the master public key and the public identity of the *prover*, and then sends only the cipher-text to the *prover*. The *prover* decapsulates the cipher text and returns the result as a response to the *verifier*. The *verifier* checks whether the response received from the *prover* is same as the initial random string he used. This way of identification seems to be very easy and simple and it defines our generic framework in the identity-based model. Following the generic construction, we independently propose a novel IBI scheme based on the q-BDHI hard problem assumption and then prove its security in the random oracle model.

Paper Organization. The paper is organized as follows: In Section 2 we fix some notations and discuss the various preliminaries used in the subsequent sections of the paper. In Section 3 we review the formal definitions and security models of IBI and ID-KEM schemes. In Section 4 we give the generic construction for deriving a new IBI scheme from an OW-CCA2 secure ID-KEM. Section 5 covers the construction of the new two-round IBI scheme in the random oracle model and its security proof in the random oracle model. We compare the efficiency of our scheme with the existing schemes in Section 6. Finally we conclude our work in Section 7.

2 Preliminaries

In this section, we discuss the primitives required, including bilinear pairings, their properties and some of the associated hard problems.

2.1 Bilinear Groups and Assumptions

The scheme proposed will require groups equipped with a bilinear map. We review the necessary facts about bilinear maps and associated groups in a similar fashion as that given in Boneh et al [8].

- $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are multiplicative cyclic groups of prime order p.
- u_1 is a generator of $\mathbb{G}_1$ and u_2 is a generator of $\mathbb{G}_2$.
- ϕ is an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ with $u_1 = \phi(u_2)$.
- $\hat{e}$ is a map, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

The bilinear map $\hat{e}$ must satisfy the following properties:

- *Bilinearity*: For all $u \in \mathbb{G}_1$, all $v \in \mathbb{G}_2$ and all $a,b \in \mathbb{Z}$ we have $\hat{e}(u^a,v^b) = \hat{e}(u,v)^{ab}$.
- *Non-degeneracy*: $\hat{e}(u_1,u_2) \neq 1$.
- *Computable*: There is an efficient algorithm to compute $\hat{e}(u,v)$, for all $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$.

There are many hard problems studied pertaining to bilinear maps. The following are some of the hard problems associated with bilinear maps which we will use in proving the security of our scheme.

Definition 2.1. ***(Computational Bilinear Diffie-Hellman (BDH))*** *problem [9] - Given group elements (g_1, g_2, g_2^x, g_2^y, g_2^z) for $x, y, z \in_R \mathbb{Z}_P$, compute $\hat{e}(g_1, g_2)^{xyz}$.*

Definition 2.2. ***(q-Bilinear Diffie-Hellman inversion (q-BDHI))*** *problem [10] - Given the group elements (g_1,g_2,g_2^x,$g_2^{x^2}$,$g_2^{x^3}$,.....,$g_2^{x^q}$) with $x \in_R \mathbb{Z}_P$, compute $\hat{e}(g_1, g_2)^{1/x}$.*

3 Formal Definition and Security Models

In this section, we formally describe IBI and ID-KEM along with their security models.

3.1 Identity Based Identification(IBI) Scheme

An IBI system consists of four Probabilistic Polynomial Time(PPT in short) algorithms (*Setup 'S'*, *Extract 'E'*, *Prove 'P'* and *Verify 'V'*) [7,6] where,

1. *Setup(S)*: The trusted key issuing authority(PKG) takes as input 1^t, where t is the security parameter, and generates the system parameters, the master public key M_{pk} and the master secret key M_{sk}. It publishes the system parameters and M_{pk} while keeps the secret M_{sk} to itself.
2. *Extract(E)*: For a user A, the PKG takes in the public identity of the user $ID_A \in \{0,1\}^*$, M_{sk} and returns the corresponding user secret key (d_{ID_A}).
3. *Identification Protocol(P and V)*: The *prover* with identity ID_A, runs the interactive *Prover(P)* algorithm with initial state as the user secret key d_{ID_A}, whereas the *verifier* runs the *Verifier(V)* algorithm with initial state as the public identity of the user ID_A and the master public key M_{pk}. At the end of the protocol, the *verifier* outputs either 1 or 0(Accept/Reject).

In the random oracle model, the three algorithms - E, P and V have access to functions H whose range may depend on M_{pk}. It is required that for all $t \in \mathbb{N}$, $ID_i \in \{0,1\}^*$, $M_{sk}, M_{pk} \in S(1^t)$, the functions H with appropriate domain and range, and $d_{ID_i} \in E(M_{sk}, ID_i : H)$, the interaction between P and V is accepted with probability one.

3.2 Security Model for IBI

We consider the security notion for IBI as proposed in [11,6]. We consider three types of attacks on the *honest prover*, namely passive attack, active attack and concurrent attack. The goal of an adversary towards an *IBI* scheme is impersonation. An impersonator is considered successful, if it interacts with an *honest verifier* as a *cheating prover* and manages to convince him with a high probability that he is the actual *prover*. In this case the IBI scheme is said to be broken and hence insecure.

We describe in general three types of attacks which an impersonator can mount on IBI schemes:

1. *Passive Attack*: This type of adversary merely listens to the transcript queries between an *honest verifier* and *honest prover*, and tries to extract information from them. The adversary only taps the channel between the *honest prover* and *verifier*. It is the most easy and weak type of attack.
2. *Active attack*: In this type of attack the adversary interacts with the *prover* as a *cheating verifier* and tries to gain knowledge using extraction queries and decapsulation of various cipher texts. After he has gathered sufficient information he acts as a *cheating prover* to convince an *honest verifier*. If he succeeds in doing so, then he has successfully broken the scheme.

3. *Concurrent Attack*: This type of adversary is similar to the active adversary, but with the difference that here he can invoke multiple *prover* instances and interacts with them at the same time.

The impersonation attack between an impersonator I and the challenger is described as a two-phased attack game described below:

- **Setup.** The challenger takes the security parameter and runs the Setup algorithm 'S'. It gives the impersonator I the system parameters and master public key M_{pk} and keeps the master secret key M_{sk} to itself.
- **Phase 1**
  1. I issues polynomial number of key extraction queries for identities ID_1, ID_2,. . .,ID_q. The challenger responds to the queries by running the extract algorithm and replies by returning the corresponding private key for the public identity queried.
  2. I issues transcript queries and some identification queries on ID_j.
  3. The queries in above steps are interleaved and asked adaptively by I. Also it is assumed that I will not query the same ID_i in the identification/transcript queries for which it has already queried the algorithm E.
- **Phase 2**
  1. I outputs a challenge identity $ID^* \neq \{ID_i$ in the extraction queries$\}$ on which it wishes to impersonate. I now plays the role of a *cheating prover*, trying to convince an *honest verifier*.
  2. I can still continue to issue extract, identification and transcript queries, with the restriction that the challenge identity is not queried for a key extract.

We say that I succeeds in impersonating if it can make an *honest verifier* accept with non-negligible probability.

Definition 3.1. *We say that an IBI scheme is (T, q_1, ϵ)-secure under passive, active and concurrent attacks if for any passive/active/concurrent impersonator I who runs in time T,*

$$Pr[I \textit{ can impersonate}] < \epsilon,$$

where I can make at most q_1 key extraction queries.

We now proceed to give more formal definitions for the IBI scheme under passive, active and concurrent settings. Let $\mathcal{A}(CP, CV)$ be an adversary consisting of two PPT algorithms, CP(*cheating prover*) and CV(*cheating verifier*) and let $t \in \mathbb{N}$ be the security parameter. There are four oracles namely *Initialize oracle*(INIT), *Corrupt oracle*(CORR), *Conversation oracle*(CONV) and *Prover oracle*(PROV) as shown in Figure 1, the access to which for the adversary depends on the type of attack as depicted in Figure 2. The adversary can initialize and corrupt identities of its own choice using the INIT and CORR oracles. In case of passive attacks(pa), the adversary gets access to the CONV oracle, which

```
Oracle INIT($ID_i$)
If $ID_i \in CU \cup HU \cup AU$ then return $\perp$
$d_{ID_i} \overset{R}{\leftarrow} S(M_{sk}, ID_i)$;  $HU \leftarrow HU \cup \{ID_i\}$
Return 1

Oracle CORR($ID_i$)
If $ID_i \notin HU \backslash AU$ then return $\perp$
$CU \leftarrow CU \cup ID_i; HU \leftarrow HU \backslash \{ID_i\}$
Return $d_{ID_i}$

Oracle CONV($ID_i$)
If $ID_i \notin HU$ then return $\perp$
$(C, d) \overset{R}{\leftarrow} \mathbf{Run}[P(d_{ID_i}) \leftrightarrow V(M_{pk}, ID_i)]$
Return $C$

Oracle PROV($ID_i, s, M_{in}$)
If $ID_i \notin HU \backslash AU$ then return $\perp$
If$(ID_i, s) \notin PS$ then
If $atk$ = $\textit{Active Attack}$ then $PS \leftarrow (ID_i, s)$
If $atk$ = $\textit{Concurrent Attack}$ then $PS \leftarrow PS \cup (ID_i, s)$
Pick random bits $\rho$ for prover algorithm $P$
$St_P[ID_i, s] \leftarrow (d_{ID_i}, \rho)$
$(M_{out}, St_P[ID_i, s]) \leftarrow P(M_{in}, St_P[ID_i, s])$
Return $M_{out}$.
```

Fig. 1. Oracles given to an adversary attacking IBI scheme

when queried with the identity ID_i of the honest and initialized user, returns a transcript of ID_i and the *verifier*, each time using fresh random bits. When an identity is initialized, it is issued a secret key by the authority. When an (honest) identity is corrupted, its secret key is returned to the adversary. CU is the set of corrupted users, HU is the set of honest users, and AU is the set of users under attack.

In the case of active attacks(aa) or concurrent attacks(ca), the adversary gets additional access to PROV. Its arguments are an identity ID_i, a session number, and a message that the adversary, playing the role of *verifier*, sends to ID_i in its role as a *prover*. The oracle maintains state for the *prover* for each active session, but allows only one session to be active at any point if the attack is an active one, rather than a concurrent one. At the end of its execution, CV transfers its state to CP and outputs an uncorrupted identity ID^*. In the second stage, CP will try to impersonate ID^*. A point in this definition worth noting is that we have allowed CP to query the same oracles as CV. This allows CP to initialize, corrupt, interact with, or see conversations involving certain identities depending on the challenge it gets from the *verifier*. The only restriction is that CP cannot submit queries involving ID^* because otherwise impersonating ID^* would become trivial. The restrictions are all enforced by the oracles themselves.

(At the end of the first stage, ID^* is added to the set of users under attack AU and, in the case of active or concurrent attacks, removed from the set of honest users HU).

```
Experiment $\mathbf{Exp}^{imp-atk}_{IBI,\mathcal{A}}(t)$        $//atk \in pa, aa, ca$

$(M_{pk}, M_{sk}) \overset{R}{\leftarrow} S(1^t)$
$HU \leftarrow \emptyset, CU \leftarrow \emptyset, AU \leftarrow \emptyset$   //set of Honest, Corrupt, and Attacked users
$PS \leftarrow \emptyset$     // set of active prover sessions
If $atk = pa$ then let OR represent CONV, else let OR represent PROV
$(ID_j, St_{CP}) \leftarrow CV(1^t, M_{pk} : INIT, CORR, OR)$
$AU \leftarrow \{ID_j\}$; If $ID_j \in HU$ then return 0.
$(C, d) \overset{R}{\leftarrow} \mathbf{Run}[CP(St_{CP} : INIT, CORR, OR) \leftrightarrow V(M_{pk}, ID_j)]$
Return $d$.
```

Fig. 2. Experiment defining *imp-atk* security of IBI scheme

3.3 ID-KEM

An *ID-KEM* is described as a quadruple of PPT algorithms(**Setup**, **Extract**, **Encapsulate**, **Decapsulate**) [12] where, the four PPT algorithms are:

- **Setup** $\mathrm{G}_{ID-KEM}(1^t)$: Given a security parameter t, the PPT algorithm generates system parameters, the master public key M_{pk}, and the master secret key M_{sk}.
- **Extract** $\mathrm{X}_{ID-KEM}(M_{pk},M_{sk},ID_A)$: Given the system parameters M_{pk} and M_{sk} and an identity string $ID_A \in \{0,1\}^*$ for an entity A, the PPT algorithm returns the corresponding private key d_{ID_A} for A.
- **Encapsulate** $\mathrm{E}_{ID-KEM}(M_{pk},ID_A)$: Given the system parameters M_{pk} and an identifier ID_A, this PPT algorithm outputs a pair (e,c) where e is a random key in the key space K corresponding to the security parameter k and c is the encapsulation of e or the ciphertext of e.
- **Decapsulate** $\mathrm{D}_{ID-KEM}(M_{pk},d_{ID_A},c)$: Given the system parameters and M_{pk}, the *prover* uses his private key d_{ID_A} to decapsulate the cipher text c and outputs the corresponding key e' or a reject symbol $\perp$. It is required that for an *honest prover* the decapsulated key e' matches with the original random key e encapsulated by *verifier* with probability one.

3.4 Security of ID-KEM

We describe the security of ID-KEM as given by Bentahar et al [13]. Consider the following two-stage games between an adversary $\mathcal{A}(\mathcal{A}_1, \mathcal{A}_2)$ of the ID-KEM and a challenger as depicted in Figure 3.

ID-OW Adversarial Game (one-wayness property)	ID-IND Adversarial Game (indistinguishability property)
1. $(M_{pk}, M_{sk}) \leftarrow \mathrm{G}_{ID-KEM}(1^t)$ 2. $(s, ID^*) \leftarrow \mathcal{A}_1^{\mathcal{O}_{ID}}(M_{pk})$ 3. $(k, c^*) \leftarrow E_{ID-KEM}(ID^*, M_{pk})$ 4. $k' \leftarrow \mathcal{A}_2^{\mathcal{O}_{ID}}(M_{pk}, c^*, s, ID^*)$	1. $(M_{pk}, M_{sk}) \leftarrow \mathrm{G}_{ID-KEM}(1^t)$ 2. $(s, ID^*) \leftarrow \mathcal{A}_1^{\mathcal{O}_{ID}}(M_{pk})$ 3. $(k_0, c^*) \leftarrow E_{ID-KEM}(ID^*, M_{pk})$ 4. $k_1 \leftarrow \mathbb{K}_{ID-KEM}(M_{pk})$ 5. $b \leftarrow \{0,1\}$ 6. $b' \leftarrow \mathcal{A}_2^{\mathcal{O}_{ID}}(M_{pk}, c^*, s, ID^*, k_b)$

Fig. 3. Two-stage game between adversary and challenger for ID-KEM

In the above games ID-OW denotes one-wayness property in the identity-based setting, ID-IND denotes indistinguishability property in the identity-based setting, s is some state information and $\mathcal{O}_{ID}$ represents the oracles to which the adversary has access. There are two possibilities for these oracles depending on the attack model for our game. The two possibilities being:

- CPA Model - In this model the adversary only has access to a private key extraction oracle which on input of $ID_i \neq ID^*$ will output the corresponding value of d_{ID_i}.
- CCA2 Model - In this model the adversary has access to the private key extraction oracle as above, but it also has access to a decapsulation oracle with respect to any identity ID_i of the adversary's choosing. The adversary has access to this decapsulation oracle, subject to the restriction that in the second phase $\mathcal{A}$ is not allowed to call $\mathcal{O}_{ID}$ with the pair (c^*, ID^*).

For a security parameter t, the adversary's advantage in the first game is defined to be:

$$\boldsymbol{Adv}_{\mathcal{A},ID-KEM}^{ID-OW-MOD}(t) = \Pr[k' = k]$$

While the advantage in the second game is given by:

$$\boldsymbol{Adv}_{\mathcal{A},ID-KEM}^{ID-OW-MOD}(t) = |Pr[b' = b] - 1|.$$

An ID-KEM is considered to be secure, in a given goal and attack model(for example, OW-CCA2(one-way CCA2)), if for all PPT adversaries $\mathcal{A}$, the advantage in the above depicted relevant game is a negligible function of the security parameter t.

4 IBI Scheme from ID-KEM

In this section we describe our generic framework to obtain an $IBI(S, E, P, V)$ scheme which is secure against concurrent attacks from an OW-CCA2 secure *ID-KEM*(G_{ID-KEM},X_{ID-KEM},E_{ID-KEM},D_{ID-KEM}) scheme, the definition of which is discussed in Section 3.3. The framework is described below:

- *Setup(S)*: PPT Master Key Generation algorithm which takes as input the security parameter(1^t) and outputs M_{pk} and M_{sk}.
- *Extract(E)*: Takes as input the public identity of a user A(ID_A), M_{sk}, M_{pk} and returns the user secret key U_{sk}.
- *Interaction Protocol(P and V)*:
  - *Verifier V*: Given ID_A, M_{pk} as input, invokes *Encapsulate* algorithm (E_{ID-KEM}) and gets (c,k), where k is a key from the Key Space and c is its corresponding cipher text.
    V sends c to *prover P*.
  - *Prover P*: Having M_{pk}, U_{sk}, c, ID_A as input, invokes *Decapsulate* algorithm (D_{ID-KEM}) and outputs $\hat{k}$.
    P sends $\hat{k}$ to V.
  - V on receiving $\hat{k}$, checks whether $k \overset{?}{=} \hat{k}$.
    If YES, then *Accept*,
    else *Reject*.

Theorem 4.1. *If an ID-KEM is OW-CCA2 secure, then the derived IBI scheme is secure against active and concurrent attacks. In other words, for an impersonator $\mathcal{I}$ that attacks the IBI scheme in the active and concurrent attack settings, there exists a PPT adversary $\mathcal{B}$ which can attack the ID-KEM in OW-CCA2 setting satisfying,*

$$\boldsymbol{Adv}_{\mathcal{I},IBI}^{imp-catk}(t) \leq \boldsymbol{Adv}_{\mathcal{B},ID-KEM}^{ID-OW-CCA2}(t)$$

where t is the security parameter.

Proof. Let ID-KEM* be an OW-CCA2 secure ID-KEM and let IBI* be the derived IBI scheme, derived using the construction stated above. Let $\mathcal{I}$ be an impersonator for IBI*. Using $\mathcal{I}$ as a subroutine, we can construct a *PPT* OW-CCA2 adversary $\mathcal{B}$, that attacks ID-KEM*. But as the ID-KEM* considered is already proven to be secure, no such adversary $\mathcal{B}$ can exist and hence our derived IBI* will also be secure. $\mathcal{B}$ plays a game with $\mathcal{I}$ acting as a *verifier* at one time and acting as decapsulation algorithm at the other time. The adversary $\mathcal{B}$ employs $\mathcal{I}$ to break the ID-KEM* as depicted in Figure 4.

$\mathcal{B}$ initializes its inner state after getting the inputs ID_A and M_{pk}, chooses a challenge ψ^* and invokes $\mathcal{I}$ on inputs ID_A and M_{pk}. $\mathcal{B}$ now proceeds as:

ACTING AS DECAPSULATION ALGORITHM - In case $\mathcal{I}$ sends a challenge message ψ to a *prover* clone, then $\mathcal{B}$ checks if $\psi = \psi^*$. If so, then $\mathcal{B}$ puts $k = \perp$ and the game is aborted, else $\mathcal{B}$ invokes its decapsulation oracle DEC(*sk*, .) for the answer of the cipher text ψ and gets as output k which it sends to $\mathcal{I}$.

ACTING AS A VERIFIER - $\mathcal{B}$ sends ψ^* to $\mathcal{I}$ as a challenge. In case $\mathcal{I}$ sends the response message $\hat{k}$ to the *verifier* V(ID_A), $\mathcal{B}$ takes it and returns $\hat{k}$ as the answer for the challenge cipher text ψ^* of ID-KEM*. Thus $\mathcal{B}$ breaks ID-KEM*.

The view of $\mathcal{I}$ in $\mathcal{B}$ is the same as its real view except for the case when $\mathcal{I}$ sends ψ=ψ^* to $\mathcal{B}$. When $\mathcal{I}$ sends ψ^* to $\mathcal{B}$, it is like relaying the transcript of its interaction between $\mathcal{I}$ and *prover* $P(sk)$ to $V(pk, ID_A)$, because the *prover* is deterministic. So $\mathcal{B}$'s response of $k = \perp$ is appropriate.

The calculation for probability follows from the fact that whenever $\mathcal{I}$ wins, $\mathcal{B}$ also wins and hence the inequality holds.

Given M_{pk} and ID_A as the input, challenger $\mathcal{B}$ acts as:

Initial Setting

- Initialize its inner state.
- Invoke $\mathcal{I}$ on inputs ID_A, M_{pk}.

Answering $\mathcal{I}$'s Queries

- Extraction Queries: $\mathcal{B}$ runs its extract algorithm $\mathrm{X}_{ID-KEM^*}(M_{pk}, M_{sk}, ID_i)$ to generate the secret key for the entity ID_i, such that $i \neq I$. $\mathcal{I}$ can ask for polynomial number of extract queries.
- In case $\mathcal{I}$ sends ψ to *prover* clone $P(d_{ID_i})$.
  - If ψ=ψ^*, then put $k = \perp$.
  - Else query decapsulation oracle(DEC) for the answer of ψ, i.e. $k \longleftarrow \mathrm{D}_{ID-KEM^*}(M_{pk}, d_{ID_A}, c)$.
  - Send k to $\mathcal{I}$.
- If $\mathcal{I}$ queries $V(ID_A, M_{pk})$ for the challenge message.
  - Send ψ^* to $\mathcal{I}$.
- In case $\mathcal{I}$ sends $\hat{k^*}$ to $V(ID_A, M_{pk})$
  - Return $\hat{k^*}$ as the answer for ψ^*.

Fig. 4. An IBI scheme from an OW-CCA2 secure ID-KEM

5 Proposed IBI Scheme

We now independently propose an IBI(S,E,P,V) scheme which is derived from the ID-KEM scheme [14] after making considerable modifications. This scheme is secure against active and concurrent attacks. The construction of the scheme is outlined below:

Let t be the security parameter. The system parameters are groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, which are all multiplicative cyclic groups of prime order p, where $p \approx 2^t$.

Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ be a bilinear map that satisfies the properties specified in Section 2.1. u_1 is generator of $\mathbb{G}_1$ and u_2 is generator of $\mathbb{G}_2$ such that $u_1 = \phi(u_2)$. The scheme uses three hash functions:

$H_1 : \{0,1\}^* \longrightarrow \mathbb{Z}_p$, H_2: $\mathbb{G}_T \longrightarrow \{0,1\}^n$, where $\{0,1\}^n$ is the message space and,

$$H_3 : \{0,1\}^* \times \{0,1\}^n \longrightarrow \mathbb{Z}_p.$$

The scheme is as follows:

- $S(1^t)$
  - Select $s \in_R \mathbb{Z}_p$.
  - Calculate $\mathrm{R} = u_1^s$.
    s is the Master Secret Key(M_{sk}).
    The system parameters along with R forms the Master Public Key(M_{pk}).

- $E(s, M_{pk}, ID_A)$
  - Compute User secret key(U_{sk}) for the identity ID_A as:
    $D_{ID_A} = u_2^{\frac{1}{(s+H_1(ID_A))}}$

- Interaction Protocol(P and V)
  - The *verifier* $V(ID_A,\ M_{pk})$ performs:

    (a) Select $k \in_R \{0,1\}^n$ i.e. randomly select a key from KEY SPACE.
    where n is of considerable bit length $\approx t$.
    (b) Select $r \longleftarrow H_3(ID_A, k)$.
    (c) Compute $Q \longleftarrow \mathrm{R}{\cdot}u_1^{H_1(ID_A)}$.
    (d) Compute $U \longleftarrow Q^r$.
    (e) Compute $V \longleftarrow k \oplus \mathrm{H}_2(\hat{e}(u_1,u_2)^r)$.
    (f) Set $c \longleftarrow (U, V)$.
    (g) Send c to *Prover P*.

  - *Prover* $P(M_{pk}, D_{ID_A}, ID_A, c)$.
    The *prover* performs the following steps:

    (a) Parse c as (U, V).
    (b) Compute $Q \longleftarrow \mathrm{R}{\cdot}u_1^{H_1(ID_A)}$.
    (c) Compute $\alpha \longleftarrow \hat{e}(U, D_{ID_A})$.
    (d) Compute $k' \longleftarrow \mathrm{H}_2(\alpha) \oplus V$, and $r \longleftarrow H_3(ID_A, k')$.
    (e) Check if $(\alpha = \hat{e}(u_1,u_2)^r)$;
    If YES, then U is CONSISTENT,
    Else ABORT.
    (f) Send k' to *verifier*.

- *Verifier* on receiving k', checks whether k=k';
  If YES, then *Accept*,
  Else *Reject*.

5.1 Security Proof

The proof of security for our scheme is explained in the Appendix A.

6 Comparison with Other Schemes

In this section, we compare our scheme with various existing IBI schemes in both standard model and random oracle model which are of Challenge-and-Response or Σ-protocol type. Our scheme, to the best of our knowledge is the first concrete IBI scheme based on the Challenge-and-Response system.

Table 1. Comparison between various IBI schemes

Schemes	Computation	Communication	Hard Problem	Model
KH-IBI [15]	$12C_M + 12C_E + 6C_P$	$5\|\mathbb{Z}_p\| + 4\|G\|$	q-SDH	Stnd
CHG-IBI [7]	$(n{+}4)C_M + 5C_E + 3C_P$	$\|\mathbb{Z}_p\| + 4\|G\|$	OMCDH	Stnd
BNN-IBI [6]	$4C_M + 7C_E + 2C_A$	$2\|\mathbb{Z}_p\| + 3\|G\|$	OMDL	RO
OkDL-IBI [6]	$8C_M + 9C_E + 4C_A$	$3\|\mathbb{Z}_p\| + 3\|G\|$	DL	RO
CCSR-IDKEM [14]	$2C_M + 6C_E + 3C_P$	$3\|G\|$	q-BDHI	RO
Our scheme	$2C_M + 3C_E + 2C_P$	$3\|G\|$	q-BDHI	RO

In Table 1, C_E, C_P, C_M, C_A represent the computational costs of - group exponential operation, bilinear group pairing operation, bilinear group multiplicative operation and group addition operation respectively. Also, in the table the terms q-SDH, OMCDH, OMDL, DL stand for q-Strong Diffie-Hellman [15], One-More Computational Diffie-Hellman [7], One-More Discrete Logarithm [6] and Discrete Logarithm [6] respectively. Also, the terms Stnd stands for Standard and RO stands for Random Oracle under the Model column heading. For better security we consider n(the size of the KEY SPACE) to be of considerable length $\approx$ security parameter t, hence $2n + |G| \approx 3|G|$. We have compared the efficiency of our scheme with the IBI scheme proposed in [15], the IBI scheme proposed in [7], BNN-IBI and OkDL-IBI schemes [6] and the ID-KEM scheme proposed in [14].

Moreover, our scheme requires only two pairing operations as one of the pairing operations in *prover*'s side of our scheme is needed only once. Thus our scheme is efficient in terms of both computation and communication as compared to other schemes making it suitable for mobile devices.

7 Conclusions

We presented a generic framework to construct IBI scheme from OW-CCA2 secure ID-KEM. This generic construction can be applied to any existing OW-CCA2 secure ID-KEM to get an IBI scheme. Further more, we proposed a new concrete IBI scheme based on ID-KEM [14] after making significant modifications and proved its security by directly reducing it to the q-Bilinear Diffie Hellman Inversion(q-BDHI) hard problem in the random oracle model. Our scheme is very efficient in terms of computation and communication complexity and hence can be used in mobile devices and smart cards where memory and efficient computation are of great importance. It still remains an open problem to construct an efficient IBI scheme that is provably secure against active and concurrent attacks using a weaker assumption in the standard model.

References

1. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, STOC 1985, pp. 291–304. ACM, New York (1985)
2. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
3. Anada, H., Arita, S.: Identification schemes from key encapsulation mechanisms. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 59–76. Springer, Heidelberg (2011)
4. Fiat, A., Shamir, A.: How To Prove Yourself: Practical Solutions to Identification and Signature Problems, pp. 186–194. Springer (1987)
5. Guillou, L.C., Quisquater, J.-J.: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 123–128. Springer, Heidelberg (1988)
6. Bellare, M., Namprempre, C., Neven, G.: Security Proofs for Identity-Based Identification and Signature Schemes. J. Cryptol. 22(1), 1–61 (2008)
7. Chin, J.-J., Heng, S.-H., Goi, B.-M.: An Efficient and Provable Secure Identity-Based Identification Scheme in the Standard Model. In: Mjølsnes, S.F., Mauw, S., Katsikas, S.K. (eds.) EuroPKI 2008. LNCS, vol. 5057, pp. 60–73. Springer, Heidelberg (2008)
8. Boneh, D., Lynn, B., Shacham, H.: Short Signatures from the Weil Pairing. J. Cryptology 17(4), 297–319 (2004)
9. Boneh, D., Franklin, M.K.: Identity-Based Encryption from the Weil Pairing. SIAM J. Comput. 32(3), 586–615 (2003)
10. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
11. Kurosawa, K., Heng, S.-H.: From Digital Signature to ID-based Identification/Signature. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 248–261. Springer, Heidelberg (2004)
12. Cheng, Z.: Simple SK-ID-KEM1 (2005)
13. Bentahar, K., Farshim, P., Malone-Lee, J., Smart, N.P.: Generic Constructions of Identity-Based and Certificateless KEMs. J. Cryptol. 21(2), 178–199 (2008)
14. Chen, L., Cheng, Z., Smart, N.P., Road, F.: An Efficient ID-KEM based on the Sakai-Kasahara key construction. In: IEE Proceedings of Information Security (2006)
15. Kurosawa, K., Heng, S.-H.: Identity-Based Identification Without Random Oracles. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3481, pp. 603–613. Springer, Heidelberg (2005)

A Proof of Security

We follow the same approach as used by Chen et al [14] to proceed with the proof. Let q_1, q_2, q_x be the number of queries that an impersonator $\mathcal{I}$ can make to H_1, H_2 and to the key extraction oracle respectively. To prove our scheme to be secure, we show that if there exists an impersonator $\mathcal{I}$ for the IBI scheme, then we can construct another algorithm $\mathcal{B}$ to solve the q-BDHI problem, where q = q_1 + q_x + 1. This construction involves $\mathcal{B}$ playing the role of challenger which will simulate the protocol.

$\mathcal{B}$ takes as input $(g_1,g_2,g_2^x,g_2^{x^2},g_2^{x^3},\ldots\ldots,g_2^{x^q}) \in \mathbb{G}_1 \times \mathbb{G}_2^{q+1}$ with $g_1 = \phi(g_2)$ and then selects an integer $I \in$ {1,......, q}. It uses these to set up the domain parameters and keys for the IBI algorithm as described below:

Algorithm $\mathcal{B}$ selects $h_0,h_1,\ldots\ldots,h_{q-1}$ uniformly at random from $\mathbb{Z}_p$. We define the event GUESS to be that in which $h_i = -x$ for some i in {1,......, q-1}.(This event can be checked by computing $g_2^{-h_i}$ for i in {1,......, q-1}, and comparing these values with g_2^x).

We say that $\mathcal{I}$ wins if it outputs the correct value of the encrypted key which is randomly selected from the KEY SPACE in its attack. The advantage with which an impersonator $\mathcal{I}$ can successfully impersonate the *prover* is represented as $Adv^{IBI}(\mathcal{I})$. By definition,

$$\begin{aligned} Adv^{IBI}(\mathcal{I}) &= Pr[\mathcal{I}\,wins \wedge GUESS] + Pr[\mathcal{I}\,wins \wedge \neg GUESS] \\ &\leq Pr[GUESS] + Pr[\mathcal{I}\,wins \mid \neg GUESS] \\ &\leq Adv^{q-BDHI}(\mathcal{B}) + Pr[\mathcal{I}\,wins \mid \neg GUESS] \qquad (1) \end{aligned}$$

The above equation follows from the fact that in the event GUESS, the algorithm $\mathcal{B}$ itself finds x which it can use to solve the q-BDHI problem by directly computing $\hat{e}(g_1, g_2)^{1/x}$.

We now describe the non trivial part of the simulation. In the remaining part of the proof, we assume that the event ¬GUESS has occurred and so all the probabilities are conditioned on this event. $\mathcal{B}$ defines the following polynomial.

Let $f(z)$ be a polynomial, where $f(z) = \prod_{i=1}^{q-1}(z + h_i)$,

Rewriting f as, $f(z) = \sum_{i=0}^{q-1} c_i \mathrm{z}^i$

The constant term c_0 is non-zero because $h_i \neq 0$ and c_i's are computable from h_i's.

$\mathcal{B}$ now computes

$$u_2 = \prod_{i=1}^{q-1}(g_2^{x^i})^{c_i} = (g_2^{f(x)}),$$

and

$$u_2' = \prod_{i=1}^{q-1} (g_2^{x^{i+1}})^{c_i} = (g_2^{x \cdot f(x)}) = u_2^x$$

Note that in the event ¬GUESS, we have $u_2 \neq 1$ and so u_2 is a generator of $\mathbb{G}_2$. Algorithm $\mathcal{B}$ also defines the polynomials,

$$f_i(z) = f(z)/(z + h_i) = \sum_{i=0}^{q-2} d_{i,j} z^j, \text{ for } 1 \leq i < q$$

Also,

$$u_2^{1/(x+h_i)} = g_2^{f_i(x)} = \prod_{j=1}^{q-2} (g_2^{x_j})^{d_{i,j}}$$

Let PS denote the set,

$$PS = \left\{ \left(h_j + h_0, u_2^{1/(x+h_i)} \right) \right\}_{j=1}^{q-1}$$

Algorithm $\mathcal{B}$ also sets,

$$t' = \prod_{i=1}^{q-1} (g_2^{x_{i-1}})^{c_i} = g_2^{(f(x)-c_0)/x}, \text{ and sets } \gamma_0 = \hat{e}(\phi(t'), u_2 \cdot g_2^{c_0})$$

It defines u_1=$\phi(u_2)$ and computes the public key of the Trusted Authority(TA) as

$$R = \phi(u_2' \cdot u_2^{-h_0}) = \phi(u_2') \cdot u_1^{-h_0} = u_1^{x-h_0}$$

We need to check that R has the correct original distribution. Since we are conditioning on the event ¬GUESS, we know that u_2 is a generator of $\mathbb{G}_2$, which means that u_1 must be a generator of $\mathbb{G}_1$ as needed in the scheme.

Consider the following distributions associated with a generator u_1 of $\mathbb{G}_1$. Note that in the description below, $\mathcal{D}_x$ is one of a collection of distributions $\{\mathcal{D}_x\}_{x \in \mathbb{Z}_p}$ parametrized by $x \in \mathbb{Z}_p$.

$$\mathcal{D} = \{u_1^s : s \leftarrow \mathbb{Z}_p\},$$
$$\text{and, } \mathcal{D}_x = \{u_1^{x-h_0} : h_0 \leftarrow \mathbb{Z}_p\}$$

Clearly, for any $x \in \mathbb{Z}_p$, these distributions are identical. Moreover, R is chosen from $\mathcal{D}$ when the identical scheme is used in reality and R is chosen from $\mathcal{D}_x$ in our simulation(conditioned on the event ¬GUESS). So, we can conclude that R has the correct distribution.

Phase 1
Algorithm $\mathcal{B}$ now invokes the first stage of the algorithm for $\mathcal{I}$ with the domain parameters that it has constructed. In this phase, the impersonator $\mathcal{I}$ will play the role of a *cheating verifier*. $\mathcal{B}$ responds to the oracle calls made by $\mathcal{I}$ as follows:

- H_1 Query on ID_i : $\mathcal{B}$ maintains a list of tuples, the $\mathcal{H}_1$ list (ID_i, h_i, D_{ID_i}) indexed by ID_i. On input of ID_i, the ith distinct query, algorithm $\mathcal{B}$ acting as a challenger responds as follows:
  1. If $i = I$, then $\mathcal{B}$ responds with h_0 and adds $(ID_i, h_0, \perp)$ to the $\mathcal{H}_1$ list.
  2. Otherwise, it selects a random element $(h_i + h_0, u_2^{1/(x+h_i)})$ from PS list(without replacement). It adds $(ID_i, h_i + h_0, u_2^{1/(x+h_i)})$ to the $\mathcal{H}_1$ list, and returns $h_i + h_0$.

  If the impersonator $\mathcal{I}$ queries the same ID_i, then $\mathcal{B}$ responds with the same result that it gave the first time by looking it up on the list.
- H_2 Query on α: $\mathcal{B}$ maintains a list $\mathcal{H}_2$ of tuples(α, β). If α appears in the list $\mathcal{H}_2$, then $\mathcal{B}$ responds with β. Otherwise it chooses β at random from $\{0,1\}^n$ and it adds (α, β) to the $\mathcal{H}_2$ list before responding with β.
- H_3 Query: Algorithm $\mathcal{B}$ generates a random value $r' \in_R \mathbb{Z}_p$ for every (ID_i, k_i) and stores this value in $\mathcal{H}_3$ list. The $\mathcal{H}_3$ list is a three tuple,(ID_i, r', k_i) and is indexed by ID_i.
- *Extraction Query* for ID_i: If ID_i does not appear on the $\mathcal{H}_1$ list then $\mathcal{B}$ first makes an H_1 query. Algorithm $\mathcal{B}$ then checks whether the corresponding value of D_{ID_i} is $\perp$. If so, then it terminates (this event corresponds to challenger $\mathcal{B}$ failing to correctly guess at what point the impersonator $\mathcal{I}$ would query $\mathcal{H}_1$ with its chosen ID^*). Otherwise, it responds with D_{ID_i}, where (ID_i, h_i, D_{ID_i}) is the entry corresponding to ID_i in the $\mathcal{H}_1$ list.
- *Impersonation Queries*: $\mathcal{B}$ will respond to these queries as:
  1. When $ID_i \neq ID_I$: $\mathcal{B}$ invokes the *Extraction Query* and then uses valid private key to interact with the impersonator $\mathcal{I}$.
  2. If $ID_i = ID_I$, $\mathcal{B}$ has to find the value of r in $\mathcal{H}_3$ list for the ID_i index. $\mathcal{B}$ searches the $\mathcal{H}_3$ list with ID_i index, so as to satisfy the condition $Q^r = U$ required in the scheme. $\mathcal{B}$ then returns the corresponding k for the r, as the response to the query, from $\mathcal{H}_3$ list. If such an r is not found, then $\mathcal{B}$ returns INVALID and continues.

$\mathcal{I}$ could guess the correct or a consistent (U, V) pair even when it has not queried the H_3 oracle with probability $1/p$. We call this event as GUESS2. In GUESS2 event, the *Impersonation Queries* will be answered as invalid though they are valid, and thus $\mathcal{I}$ will come to know that he is playing with a simulated challenger, and not in a real scenario. So $\mathcal{I}$ will stop the game with probability $1/p$ which is negligibly small.

Phase 2
After some time, $\mathcal{I}$ will terminate its first phase and will return the challenge identity ID^*. If $\mathcal{I}$ has not called H_1 with input ID^*, then $\mathcal{B}$ does so for it. The corresponding value of D_{ID^*} must be $\perp$, or else $\mathcal{B}$ will have to abort.

Algorithm $\mathcal{B}$ chooses a random value of $r \in \mathbb{Z}_p$ and a random value V^* in $\{0,1\}^*$. It computes $U^* = u_1^r$ and sets the challenge cipher text as,

$$c^* = (U^*, V^*)$$

This cipher text is now passed to $\mathcal{I}$'s second stage. $\mathcal{I}$ will continue to ask *Extraction Queries* and owing to the rules of the game, $\mathcal{B}$ will not terminate unexpectedly and will continue returning appropriate values.
At some point, the algorithm $\mathcal{I}$ acting as *cheating prover* outputs the value of the underlying key k'. For a genuine key we should have

$$k' = V^* \oplus \mathcal{H}_2(\hat{e}(U^*,D_{ID^*})).$$

If H_2 is modelled as a random oracle, we know that $\mathcal{I}$ has advantage of returning the valid k' only if the list $\mathcal{H}_2$ contains an input value

$$\alpha^* = \hat{e}(U^*,D_{ID^*}).$$

Algorithm $\mathcal{B}$ selects a value α at random from the $\mathcal{H}_2$ list and we assume that it correctly selects $\alpha = \alpha^*$, thus this adds for an additional factor of $1/\mathrm{q}_2$ to our subsequent analysis. Acting challenger $\mathcal{B}$ sets

$$\gamma = {\alpha^*}^{1/r}$$

We have that,

$$D_{ID^*} = u_2^{1/((x-h_0)+h_0)}$$

and so

$$\gamma = \hat{e}(u_1, u_2)^{1/x}$$

The challenger's job is to compute $\hat{e}(g_1, g_2)^{1/x}$. It computes,

$$\begin{aligned}\gamma/\gamma_0 &= \hat{e}(g_1, g_2)^{f(x)\cdot f(x)/x} \;/\; \hat{e}(g_1^{(f(x)-c_0)/x}, g_2^{f(x)+c_0}) \\ &= \hat{e}(g_1, g_2)^{(f(x)\cdot f(x)/x)-(f(x)\cdot f(x)/x)+(c_0^2/x)} \\ &= \hat{e}(g_1, g_2)^{c_0^2/x}.\end{aligned}$$

and $\mathcal{B}$ solves the q-BDHI problem by outputting $\hat{e}(g_1, g_2)^{1/x} = (\gamma/\gamma_0)^{1/c_0^2}$.

The above procedure for calculating the solution can fail if: (1)$r = 0$, (2)c_0 =0. However, this will not happen if $h_i \neq 0$ for i=0,.....,q-1 and $r \neq 0$. We say that the event FAIL occurs if at least one of these condition fails. We have,

$$\begin{aligned} Pr[\mathcal{I}\ wins \mid \neg GUESS] &= Pr[\mathcal{I}\ wins \wedge \neg FAIL \mid \neg GUESS] + \\ &\qquad Pr[\mathcal{I}\ wins \wedge FAIL \mid \neg GUESS] \\ &\leq Pr[\mathcal{I}\ wins \mid \neg FAIL \wedge \neg GUESS] + \frac{q+1}{p} \\ &\leq Pr[\mathcal{I}\ wins \mid \neg FAIL \wedge \neg GUESS] + \frac{q+1}{p} \end{aligned} \tag{2}$$

Let us denote the event that $\mathcal{I}$ makes the query α^* during its attack by ASK.

$$\begin{aligned} Pr[\mathcal{I}\ wins \mid \neg GUESS \wedge \neg FAIL] &= Pr[\mathcal{I}\ wins \wedge ASK \mid \neg GUESS \wedge \neg FAIL] + \\ &\qquad Pr[\mathcal{I}\ wins \wedge \neg ASK \mid \neg GUESS \wedge \neg FAIL] \\ &= Pr[\mathcal{I}\ wins \wedge ASK \mid \neg GUESS \wedge \neg FAIL] \\ &\quad + \frac{1}{2^n} \end{aligned} \tag{3}$$

The last inequality follows from the fact that in a random oracle model, if the event ASK does not occur, then $\mathcal{I}$ has no information about the message encrypted in the challenge ciphertext.

To conclude the proof we note that when the event ASK happens, then $\mathcal{B}$ succeeds in solving q-BDHI problem if,

(1) $\mathcal{B}$ picks the correct index I, which happens with probability $1/(q_1 + q_x + 1)$, and

(2) $\mathcal{B}$ chooses the correct entry α^* from list H_2, which happens with probability $1/q_2$,

Thus, we have,

$$Adv^{q-BDHI}(\mathcal{B}) \geq \left(\frac{1}{q_1 + q_x + 1}\right) \cdot \left(\frac{1}{q_2}\right) \cdot Pr[\mathcal{I} \text{ wins} \wedge \text{ASK} \mid \neg GUESS \wedge \neg FAIL]$$

$$Adv^{q-BDHI}(\mathcal{B}) \geq \frac{Pr[\mathcal{I}\ wins \wedge ASK \mid \neg GUESS \wedge \neg FAIL]}{((q_1 + q_x + 1) \cdot q_2)}$$

$$Adv^{q-BDHI}(\mathcal{B}) \geq \frac{Pr[\mathcal{I}\ wins \mid \neg GUESS \wedge \neg FAIL]}{((q_1 + q_x + 1) \cdot q_2)} - \frac{1}{(2^n \cdot ((q_1 + q_x + 1) \cdot q_2))} \qquad [Using (3)]$$

$$Adv^{q-BDHI}(\mathcal{B}) \geq \frac{Pr[\mathcal{I}\ wins \mid \neg GUESS]}{((q_1 + q_x + 1) \cdot q_2)} - \frac{1}{(2^n \cdot ((q_1 + q_x + 1) \cdot q_2))} - \frac{(q+1)}{(p \cdot (q_1 + q_x + 1) \cdot q_2)} \qquad [Using (2)]$$

$$Adv^{q-BDHI}(\mathcal{B}) \cdot ((q_1 + q_x + 1) \cdot q_2) \geq Pr[\mathcal{I}\ wins \mid \neg GUESS] - \frac{1}{2^n} - \frac{(q+1)}{p}$$

$$Adv^{q-BDHI}(\mathcal{B}) \cdot ((q_1 + q_x + 1) \cdot q_2) + \frac{1}{2^n} + \frac{(q+1)}{p} \geq Pr[\mathcal{I}\ wins \mid \neg GUESS] \tag{4}$$

Putting (4) in (1), we get:

$$Adv^{IBI}(\mathcal{I}\ wins) \leq Adv^{q-BDHI}(\mathcal{B}) + Adv^{q-BDHI}(\mathcal{B}) \cdot ((q_1 + q_x + 1) \cdot q_2) + \frac{q+1}{p} + \frac{1}{2^n}$$

$$Adv^{IBI}(\mathcal{I}\ wins) \leq ((q_1 + q_x + 1) \cdot q_2) + 1) \cdot Adv^{q-BDHI}(\mathcal{B}) + \frac{q+1}{p} + \frac{1}{2^n}$$

$$Adv^{IBI}(\mathcal{I}\ wins) - \left(\frac{q+1}{p} + \frac{1}{2^n}\right) \leq ((q_1 + q_x + 1) \cdot q_2 + 1) \cdot Adv^{q-BDHI}(\mathcal{B})$$

$$Adv^{q-BDHI}(\mathcal{B}) \geq \frac{Adv^{IBI}(\mathcal{I}\ wins) - (\frac{q+1}{p} + \frac{1}{2^n})}{((q_1 + q_x + 1) \cdot q_2 + 1)}$$

Since $Adv^{IBI}(\mathcal{I}\ wins)$ is a non-negligible quantity and quantities $\frac{q+1}{p}$ and $\frac{1}{2^n}$ are negligibly small, we can easily infer that $Adv^{q-BDHI}(\mathcal{B})$ is non-negligible as q_1, q_x, q_2 are polynomial quantities. Hence, the challenger solves the q-BDHI problem with non-negligible probability, which is not possible and so our scheme holds secure.