# Efficient Adaptively Secure IBBE from the SXDH Assumption

Somindu C. Ramanna and Palash Sarkar
Applied Statistics Unit
Indian Statistical Institute
Kolkata, India.
E-mail: {somindu_r,palash}@isical.ac.in

**Abstract**

This paper describes the first constructions of identity-based broadcast encryption (IBBE) using Type-3 pairings which can be proved secure against adaptive-identity attacks based on the SXDH assumption (which is a static, if not a standard, assumption) achieving a security degradation which is not exponential in the size of the target identity set. The constructions are obtained by extending the currently known most efficient identity-based encryption scheme proposed by Jutla and Roy in 2013. The new constructions fill both a practical and a theoretical gap in the literature on efficient IBBE schemes.

**Keywords:** broadcast encryption, identity-based broadcast encryption, Type-3 pairings, dual-system encryption, standard assumptions.

## 1 Introduction

Broadcast encryption (BE) enables broadcasting encrypted data to a set of users so that only a subset of these users, called *privileged users*, are able to decrypt. Users who are unable to decrypt the broadcasted information are called *revoked* users. The sets of privileged and revoked users form a partition of the set of all users and these sets can vary with each broadcast. A BE system is said to be *collusion resistant* if no information of the encrypted data is leaked even if all revoked users collude. BE has a wide range of applications including pay-TV, copyright protection of digital content and encrypted file systems.

At a broad level, there are two settings for BE. In symmetric key BE, there is a centre which pre-distributes key material to the users. During a broadcast, the actual message is encrypted with a session key and the session key undergoes several encryptions using a subset of keys corresponding to the privileged users. In such a scenario, it is not possible for an entity other than the centre to broadcast an encrypted message. BE in the public key setting (PKBE) addresses this problem. Users have public and private keys. Anybody can encrypt and broadcast a message but only the intended recipients (privileged users) can decrypt.

Identity-based broadcast encryption (IBBE) is an extension of PKBE. As in the case of identity-based encryption (IBE), there is a private key generator (PKG) which issues decryption keys to entities against their identities. A message can be encrypted to a set of privileged identities. The motivation of IBBE is to reduce the communication overhead when the same message is to be sent to a group of identities. Further, anyone can broadcast given just the public parameters of the PKG. The focus of this work is the construction of IBBE schemes.

### 1.1 Issues Regarding the Construction of IBBE Schemes

There are several important issues to be considered for IBBE schemes. Below we briefly discuss some of these issues.

**Security model:** The security model for IBBE allows an adversary to specify a target set of identities such that the adversary can compromise the security of an encryption to this target set. The model also allows the adversary to corrupt entities and obtain the decryption keys corresponding to their identities with the restriction that the corrupted set of identities is disjoint from the target set of identities. Depending on when the adversary specifies the target set leads to two different security notions. The weaker notion, called selective-identity security (abbreviated as sID), requires the adversary to specify the target set *before* seeing the public parameters or corrupting any entity. The stronger notion, called adaptive-identity security (abbreviated as aID), allows the adversary to specify the target set after it has corrupted a set of identities (and also allows it to corrupt identities after specifying the target set). It is desirable to obtain schemes which are secure against adaptive-identity attacks.

**Security degradation:** In reduction proofs, the security result is quantified in terms of a security bound. Such a bound states that the advantage of breaking the scheme is upper bounded by the advantage of solving some hard problem multiplied by a factor. This factor represents the security degradation. For some IBBE schemes achieving security against adaptive-identity attacks, the degradation is exponential in the size of the target set of identities. As a consequence, the security result becomes meaningless for even moderately sized target identity set.

**Hardness assumption:** As in most public-key schemes, the proof of security of the primitive is based on the assumption that some well formulated problem is computationally hard. There is a small subset of such problems which are considered to be standard. Apart from standard hardness assumptions, designers sometimes have to create new hard problems to effect a reduction. These problems are often parametrised by a quantity arising either in the construction or the proof. If not, they are termed *static*. Since such non-standard problems are less studied, a basic theme of research is to try and obtain schemes which can be proved secure under standard and/or static assumptions.

**Header size:** In all BE schemes, the actual message undergoes a single encryption with a session key. In addition to this, the ciphertext contains some additional information which allows a privileged user to obtain the session key and recover the message. This additional information constitutes the header of the ciphertext. To reduce the communication overhead it is desirable to reduce the size of the header as much as possible. So, BE schemes with lower header sizes are preferable.

**User key size:** The amount of key material that a user has to store is an important parameter. Practical deployment may require storing such material in smart cards. Consequently, it is of interest to try and reduce the size of user keys as much as possible.

**Type of pairing:** Efficient constructions of IBBE fall in the general category of pairing-based cryptography. Such constructions require a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of some prime order $p$. Three kinds of pairings are identified in the literature: Type-1, where $\mathbb{G}_1 = \mathbb{G}_2$; Type-2, where an efficiently computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ is known; and Type-3, where there are no known efficiently computable isomorphisms from $\mathbb{G}_1$ to $\mathbb{G}_2$ or vice versa. It has been reported in the literature [SV07, GPS08, CM11], that among the different types of pairings, it is the Type-3 pairings which provide the most compact parameter sizes and the most efficient algorithms. Further, Type-1 pairings are usually defined over low characteristics fields and recent advances [BGJT14, Jou13, GKZ14a, AMORH14, GKZ14b] in algorithms for discrete log computations over such fields have raised serious question marks about the security of Type-1 pairings [Gal14]. From both efficiency and security considerations, constructions based on Type-3 pairings are desirable.

## 1.2 Our Contributions

We present the first IBBE constructions using Type-3 pairings that achieve security against adaptive-identity attacks under the Symmetric eXternal Diffie-Hellman (SXDH) assumption. All previously known IBBE schemes either used Type-1 pairings, or achieved security against selective-identity attacks, or used parametrised assumptions, or were obtained from dual pairing vector spaces making them quite inefficient.

A simple way to encrypt a single message to a set of identities is to use an IBE scheme to encrypt it separately to each of the identities. Such a strategy, however, does not allow any savings in the header size. The encryption algorithm of an IBE scheme results in a ciphertext which consist of several elements of $\mathbb{G}_1$ and/or $\mathbb{G}_2$. To obtain a non-trivial IBBE scheme, it is of interest to try and share some of the group elements in the ciphertext across all the encryptions. This will lead to a reduction in the size of the ciphertext over the trivial scheme of separate encryption to each identity.

Currently, the most efficient IBE scheme that is known is due to Jutla and Roy [JR13][1]. In this work, we investigate the possibility of converting this IBE scheme into an IBBE scheme. The intuitive idea is to share the randomiser across all the identities. Doing this directly, however, does not admit a security proof. To get around the problem, we need to put a bound on the size of the set of identities to which a single message can be simultaneously encrypted and then let the size of the public parameters be determined by this bound. The group elements in the public parameters allow the computation of polynomial hash of each of the identities. These hashes vary with the identities whereas the group elements which do not depend on the identity remain the same for all the identities. It is due to this feature that we are able to get a substantial practical reduction in the size of the ciphertext. The resulting scheme, denoted $\mathcal{IBBE}_1$, can be proved to be secure against adaptive-identity attacks using the dual-system proof technique introduced by Waters [Wat09]. The underlying hardness assumption consists of the decisional Diffie-Hellman (DDH) assumptions in the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ (DDH1 and DDH2 respectively) collectively known as the SXDH assumption.

Ciphertexts in $\mathcal{IBBE}_1$ contain $\ell$ $\mathbb{Z}_p$ elements (called tags) where $\ell$ is the number of identities to which encryption is to be done. Our second scheme, $\mathcal{IBBE}_2$, is a modification of $\mathcal{IBBE}_1$ which provides a method whereby the number of tags in the ciphertext goes down and hence results in shorter ciphertexts. The proof of security of this scheme can be reduced from the proof of security of $\mathcal{IBBE}_1$ using a hybrid argument. We use a method from [GW09] whereby the tags can be generated using a hash function resulting in an even further reduction in the size of the ciphertext. The reduction is more significant in the case of $\mathcal{IBBE}_1$ than in the case of $\mathcal{IBBE}_2$. The trade-off for doing this is that the hash function needs to be modelled as a random oracle for the security proof. User storage in both $\mathcal{IBBE}_1$ and $\mathcal{IBBE}_2$ consists of a constant number of group elements of $\mathbb{G}_2$.

Naor, Naor and Lotspiech [NNL01] had provided a combinatorial framework called the complete subtree (CS) scheme for symmetric key BE. Dodis and Fazio [DF03] had shown how to combine an IBE scheme with the CS scheme to obtain a PKBE scheme. We build on this framework and show that combining an IBBE scheme with the CS scheme leads to a PKBE scheme with even better parameters. Concretely, we discuss the issue of combining the CS scheme with $\mathcal{IBBE}_1$. A previous work [BSNS05] had proposed a particular IBBE obtained from the Boneh-Franklin IBE scheme [BF03] and had used an idea similar to ours to obtain a PKBE scheme.

**Discussion on the SXDH Assumption.** For both the DDH and the DLin problems there are no known efficient algorithms to solve these problems in a suitable subgroup of the points on an elliptic curve. The situation changes when we move to pairing groups. For Type-1 pairings, i.e., in the case $\mathbb{G}_1 = \mathbb{G}_2$, DDH becomes easy to solve, whereas DLin is still conjectured to be hard. On the other hand, for Type-3 pairings, the situation is different. As mentioned earlier, for such pairings, there are no known efficiently computable isomorphisms from $\mathbb{G}_1$ to $\mathbb{G}_2$ or from $\mathbb{G}_2$ to $\mathbb{G}_1$. A consequence of this is that the easy algorithm for solving

[1] In the scheme in [JR13] ciphertexts consist of elements of $\mathbb{G}_2$. A simple variant of this scheme has ciphertexts which consist of elements of $\mathbb{G}_1$ [RS13]. This variant is the IBE of choice.

DDH in Type-1 pairings no longer applies and for Type-3 pairings there are no known efficient algorithms to solve DDH1 or DDH2.

The DDH1 (resp. DDH2) problem would become easy if one were able to find an efficiently computable isomorphism from $\mathbb{G}_1$ to $\mathbb{G}_2$ (resp. $\mathbb{G}_2$ to $\mathbb{G}_1$). The non-existence of such isomorphisms is an underlying assumption required for the SXDH assumption to hold. Presently, the isomorphism problem has perhaps not been studied in detail and so, considering SXDH to be a standard assumption may not be universally accepted. On the other hand, we do mention that there is evidence [Ver04, GR04] that SXDH is indeed hard. Further, starting from Waters' remarks about the possible efficiency improvements of his dual-system IBE using the SXDH assumption, several schemes have been proposed whose security relies on this assumption [CLL$^+$12, CW13, JR13]. In view of the above discussion, we consider SXDH to be a 'natural' assumption which has been used earlier and has some evidence to support the assumption. In the current state of knowledge, there is no evidence to suggest that for Type-3 pairings, SXDH problem is easier than DLin.

## 1.3 Previous and Related Works

The notion of broadcast encryption was introduced by Fiat and Naor in [FN93]. They describe a symmetric key scheme that achieves bounded collusion resistance. The first fully collusion secure BE (for stateless receivers) was proposed by Naor, Naor and Lotspiech [NNL01]. They describe two symmetric key based BE constructions. Dodis and Fazio [DF02] used techniques from (hierarchical) identity-based encryption to instantiate the subset cover framework thereby leading to the first fully collusion resistant public key broadcast encryption (PKBE) schemes. The ciphertext size in their constructions is linear in the number of privileged users.

Boneh, Gentry and Waters [BGW05] proposed the first PKBE system achieving constant size ciphertexts. The scheme can be proved secure without random oracles but in the weaker selective model. Delerablée, Paillier and Pointcheval introduced dynamic broadcast encryption in [DPP07] and proposed two (partially) adaptively secure constructions. In dynamic BE schemes, a (new) user can join at any point of time. The confidentiality of a broadcasted message prior to joining of the new user must not be compromised after the join. The join operation requires that the sender is made aware of the public key corresponding to the new user.

The first adaptively secure schemes were proposed by Gentry and Waters [GW09] in both the public key and identity-based settings. They describe two kinds of schemes – one achieving security without random oracles with ciphertext size linear in the number of privileged users and the other consisting of constant size ciphertexts with security relying on the use of random oracles. More recently, the case of adaptive CCA-security was considered in [PPS11, PPSS12]. The construction proposed in the later work has the constant-size ciphertext feature while the former allows users to join the system dynamically.

All the schemes mentioned so far are secure under some non-standard and parametrised assumptions. The first BE scheme with a proof of security under static assumptions was proposed by Waters [Wat09] using the dual system encryption method. The scheme has constant size ciphertexts but the user key size is linear in the total number of users. A revocation system with constant sized keys was proposed in [LSW10] with ciphertext size growing linearly in the number of revoked users and security from static assumptions.

The concept of identity-based broadcast encryption (IBBE) was formalised by Barbosa and Farshim [BF05] and independently by Baek, Safavi-Naini and Susilo [BSNS05]. They called it multi-receiver identity-based encryption (MR-IBE). The work [BSNS05] described a pairing based construction based on the Boneh-Franklin IBE [BF03] that could be proved selectively secure in the random oracle model. A key encapsulation scheme for multiple parties obtained by extending the OR-construction of Smart [Sma04] to the identity-based setting was presented in [BF05]. Security relies on the use of random oracles.

The construction in [PKL08] (a corrected and improved version of [CS06b]) achieves a trade-off between the ciphertext size and the user key size. Ciphertexts are of size $|S|/N$, and user secret keys are of size $N$ where $N$ is a parameter of the protocol (representing the maximum number of identities that the adversary is allowed to corrupt during simulation). This was the first scheme with sub-linear sized ciphertexts.

Abdalla *et al.* [AKN07] provided a generic construction from "wicked IBE" with constant-sized ciphertexts but user storage quadratic in $m$, the maximum number of recipients of a ciphertext. Both schemes ([PKL08] and [AKN07]) are selectively secure without random oracles. In 2007, Delerablée [Del07] proposed an IBBE construction with constant size ciphertexts and secret keys. The public parameters have size $O(m)$. Security was proved in the selective identity model.

Gentry and Waters [GW09] were the first to propose adaptively secure IBBE systems achieving linear and sub-linear sized ciphertexts. However, their proofs were based on non-standard assumptions parameterised by $m$.

The issue of *anonymity* in IBBE schemes was addressed by Fan, Huang and Ho [FHH10]. Ciphertexts in anonymous BE schemes completely hide information about the set of privileged recipients. A secret key for a privileged user's identity allows decryption without knowledge of other privileged identities. The construction proposed in [FHH10] was based on Boneh-Franklin IBE but only selectively secure in the random oracle model. Later work by Ren et al. [RNZ14] proposed a similar construction based on Waters' IBE [Wat05] that achieved adaptive security. This scheme is the most efficient among fully secure schemes based on standard assumptions but the security degradation is $O(n^{\ell})$ (where identities are $n$-bit strings and $\ell$ is the number of identities in the target group) thus making the reduction meaningful only for very small values of $\ell$.

**Note:** A basic functionality of any identity-based system is that it is dynamic. The PKG should be able to generate keys for any identity from the identity space; further, an identity-based system should allow for encryption to be possible to an identity even before a key for that identity has been generated. By extension, any proper identity-based broadcast encryption scheme should also be dynamic and this is true of the schemes that we describe.

In the following section, we compare the various parameters of our constructions to that of the constructions appearing in the literature. IBBE can be viewed as a special case of inner product encryption (IPE). The constructions of adaptively secure IPE given in [LOS$^+$10, OT11, OT12] use dual pairing vector spaces and hence lead to very inefficient schemes. A construction of adaptively secure IPE is given in [AL10] which is based on Waters' IBE [Wat09].

**Pair Encoding and Attribute-Based Encryption.** Attrapadung [Att14] introduced a new primitive called *pair encoding schemes* that lead to generic constructions of fully secure predicate encryption schemes. Constructions were based on composite-order pairings and recently prime-order variants have been proposed in [Att15]. One of the implications was a fully secure attribute-based encryption (ABE) scheme with constant size ciphertexts. The ABE scheme can be specialised to obtain an IBBE system with constant sized ciphertexts but public parameter and key sizes being proportional to $m$. Furthermore, the construction is based on composite-order pairings and security is based on a non-standard parameterised assumption. In the context of IBBE, the new scheme does not offer any improvements over the Gentry-Waters scheme in terms of the underlying assumptions. We do not include this scheme in our comparison tables.

## 1.4 Comparison to Existing Schemes

Tables 1 and 2 provide comparison of $\mathcal{IBBE}_1$, $\mathcal{IBBE}_2$, $\mathcal{IBBE}_1^{\mathrm{RO}}$ and $\mathcal{IBBE}_2^{\mathrm{RO}}$ with previously known IBBE systems secure with and without random oracles respectively. The ones derived as special cases of inner-product encryption have been omitted due to reasons explained earlier. Apart from these, we have tried to include all previously known IBBE schemes appearing in the literature.

We consider the following schemes for comparison: the early selectively secure constructions [BSNS05, BF05] based on random oracles (ROs); constructions in [CS06a, CS06b] with selective security and without ROs; constant-size ciphertext IBBE schemes selectively secure (with and without ROs) proposed by Delerablée [Del07]; generic constructions of IBBE schemes from "wicked" IBE schemes by Abdalla-Kiltz-Neven [AKN07] instantiated with BBG-HIBE (without ROs) and GS-HIBE (with ROs); two adaptively secure IBBE constructions proposed by Gentry and Waters [GW09] – one with linear size (in number of privileged users) ciphertexts and the other with sub-linear size ciphertexts referred to as (a) and (b) respectively and a variant of scheme (a) based on ROs.

The basis for comparison are the following parameters – type of pairing, number of group elements in $\mathcal{PP}$ (denoted #pp) from $\mathbb{G}_1$ and $\mathbb{G}_T$, number of elements in $\mathsf{Hdr}$ (#hdr) from $\mathbb{G}_1$ and $\{0,1\}^n$ (the key space of a DEM scheme as part of the KEM-DEM framework), number of elements in a user key (#ukey) from $\mathbb{G}_2$ and $\mathbb{Z}_p$, efficiency of encryption/encapsulation (#enc) measured in terms of number of scalar multiplications in $\mathbb{G}_1$ (denoted $M_1$) and number of multiplications in $\mathbb{Z}_p$ (denoted $M_p$), number of pairings required for decryption/decapsulation (#dec), security model and computational assumptions. The quantity $m$ denotes the maximum size of the privileged users' set and $\ell$ ($\leq m$) is the size of the intended recipient set chosen during encryption. In construction [GW09]-(b) as well as scheme $\mathcal{IBBE}_2$, the maximum number of privileged users is given by $m = m_1 m_2$. The size of the set of users chosen during encryption is given by $\ell = \ell_1 \ell_2$ where $\ell_1 \leq m_1$ and $\ell_2 \leq m_2$. In [RNZ14], $n$ denotes the length of an identity. In the comparison, we ignore descriptions of hash functions, pseudorandom functions (PRFs) and other parameters that do not have any significant effect on the space-efficiency.

In the paper by Gentry and Waters [GW09], construction (b) consists of $\ell_1$ separate symmetric encryptions of the message under the $\ell_1$ keys generated by calls to the encapsulation algorithm of construction (a). In practice, the $\ell_1$ keys would be used to mask a single session key via a KDF and there would be single encryption of the message under the session key. We take this into account in the comparison tables.

'CCA' stands for chosen ciphertext attack whereas 'CPA' stands for chosen plaintext attacks. Apart from [BF05] all other schemes, including ours, have been proved secure against CPA. While CCA-security is the final desired goal, the first challenge in the design of IBBE schemes is to be able to handle adaptive-identity attacks. Most of the research works on this topic have focussed on this goal. Given that our constructions provide satisfactory solutions to the first problem, adapting known techniques to efficiently achieve CCA-security should form the focus of future work. In addition to CPA-security, the constructions in [RNZ14, FHH10] also achieve anonymity based on DBDH and DBDH-M respectively. Since we do not deal with anonymity in our constructions, this property is not included in our comparisons.

The assumptions mentioned in the tables are as follows: decisional bilinear Diffie-Hellman (DBDH), Gap bilinear DH (Gap-BDH), decisional bilinear DH exponent (DBDHE), generalised decisional DH exponent (GDDHE), DBDHE sum (DBDHES), modified DBDH (DBDH-M), security of a pseudorandom function (PRF) and SXDH. Recall that SXDH assumption is a single name for the two decisional Diffie-Hellman (DDH) assumptions in the groups $\mathbb{G}_1$ and $\mathbb{G}_2$.

| Scheme | Pairing | #pp | | #hdr | | #ukey | | #enc | | #dec | Security | Assumptions |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\mathbb{G}_1$ | $\mathbb{G}_T$ | $\mathbb{G}_1$ | $\{0,1\}^\kappa$ | $\mathbb{G}_2$ | $\mathbb{Z}_p$ | $M_1$ | $M_p$ | | | |
| [BSNS05] | Type-1 | 3 | – | $\ell+1$ | – | 1 | – | $O(\ell)$ | – | 2 | sID-CCA | Gap-BDH |
| [BF05] | Type-1 | 3 | – | $3\ell$ | – | 1 | – | $O(\ell)$ | $O(1)$ | 2 | sID-CCA | Gap-BDH |
| [AKN07] (from GS-HIBE) | Type-1 | $m+2$ | 1 | $\ell+1$ | – | $O(m)$ | – | $O(m)$ | – | $\ell+1$ | sID-CPA | DBDH |
| [Del07]-ROM | Type-1 | $m+2$ | 1 | 2 | – | 1 | 1 | $O(\ell)$ | $O(2^\ell)$ | 2 | sID-CPA | GDDHE |
| [GW09]-(a)-ROM | Type-1 | $4m+2$ | – | 4 | – | 1 | 1 | $O(\ell)$ | $O(2^\ell)$ | 2 | aID-CPA | $m$-DBDHES |
| [FHH10] | Type-1 | 3 | – | $\ell+2$ | $O(1)$ | 1 | – | $O(\ell^2)$ | $O(\ell^2)$ | 2† | sID-CPA | co-DBDH |
| $\mathcal{IBBE}_1^{\mathrm{RO}}$ | Type-3 | $m+4$ | 1 | $\ell+2$ | 1 | 5 | – | $O(m\ell)$ | $O(1)$ | 3 | aID-CPA | SXDH |
| $\mathcal{IBBE}_2^{\mathrm{RO}}$ | Type-3 | $m_2+4$ | 1 | $\ell+2\ell_1$ | $\ell_1+1$ | 5 | – | $O(m_2^2\ell_1)$ | $O(1)$ | 3 | aID-CPA | SXDH |

†: Additionally, $\ell$ map-to-point computations are required.

Table 1: Comparison of $\mathcal{IBBE}_1^{\mathrm{RO}}$ and $\mathcal{IBBE}_2^{\mathrm{RO}}$ with previously known IBBE systems in the random oracle model. In the case of Type-1 pairings, $\mathbb{G}_2$ is the same as $\mathbb{G}_1$.

Based on Tables 1 and 2, we have the following observations.

1. The new constructions use Type-3 pairings whereas the other constructions in the tables used Type-1 pairings. This leads to significantly smaller sizes for $\mathbb{G}_1$ which in turn leads to smaller ciphertexts and faster encryption and decryption algorithms.

| Scheme | Pairing | #pp | | #hdr | | | #ukey | | #enc | | #dec | Security | Assumptions |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\mathbb{G}_1$ | $\mathbb{G}_T$ | $\mathbb{G}_1$ | $\{0,1\}^\kappa$ | $\mathbb{Z}_p$ | $\mathbb{G}_2$ | $\mathbb{Z}_p$ | $M_1$ | $M_p$ | | | |
| [CS06a] | Type-1 | $m+4$ | – | $\ell+1$ | – | – | 2 | – | $O(m\ell)$ | $O(1)$ | 2 | sID-CPA | DBDH |
| [CS06b] | Type-1 | $m+4$ | – | $2\ell$ | – | – | $m+2$ | – | $O(\ell)$ | $O(1)$ | 2 | sID-CPA | $(m+1)$-DBDHE |
| [AKN07] (from BBG-HIBE) | Type-1 | $m+4$ | – | 2 | – | – | $\ell+1$ | – | $O(\ell)$ | $O(1)$ | 2 | sID-CPA | $(\ell-1)$-DBDHE |
| [Del07] | Type-1 | $m+2$ | 1 | 2 | – | – | 1 | – | $O(\ell)$ | $O(2^\ell)$ | 2 | sID-CPA | GDDHE |
| [GW09]-(a) | Type-1 | $4m+2$ | – | 4 | – | $\ell$ | 1 | 1 | $O(\ell)$ | $O(2^\ell)$ | 2 | aID-CPA | $m$-DBDHES, PRF |
| [GW09]-(b) | Type-1 | $4m_2+2$ | – | $4\ell_1$ | $\ell_1$ | $\ell_2$ | 1 | 1 | $O(\ell)$ | $O(2^{\ell_2}\ell_1)$ | 2 | aID-CPA | $m$-DBDHES, PRF |
| [RNZ14] | Type-1 | $n+3$ | 1 | $\ell+1$ | 1 | – | 2 | – | $O(\ell^2)$ | $O(\ell^2)$ | 2 | aID-CPA† | DBDH |
| [AL10] | Type-1 | $m+11$ | 1 | 9 | – | 1 | $m+6$ | $m-1$ | $12+m$ | $m+1$ | 8‡ | aID-CPA | DLIN, DBDH |
| $\mathcal{IBBE}_1$ | Type-3 | $m+4$ | 1 | $\ell+2$ | – | $\ell$ | 5 | – | $O(m\ell)$ | $O(1)$ | 3 | aID-CPA | SXDH |
| $\mathcal{IBBE}_2$ | Type-3 | $m_2+4$ | 1 | $\ell+2\ell_1$ | $\ell_1$ | $m_2$ | 5 | – | $O({m_2}^2\ell_1)$ | $O(1)$ | 3 | aID-CPA | SXDH |

†: The security degradation is exponential in the size of the target identity set.

‡: Additional $m-1$ multiplications in $M_1$ are required.

Table 2: Comparison of $\mathcal{IBBE}_1$ and $\mathcal{IBBE}_2$ with existing IBBE systems without random oracles. In the case of Type-1 pairings, $\mathbb{G}_2$ is the same as $\mathbb{G}_1$.

2. Apart from $\mathcal{IBBE}_1$, $\mathcal{IBBE}_1^{\mathrm{RO}}$, $\mathcal{IBBE}_2$, $\mathcal{IBBE}_2^{\mathrm{RO}}$; the constructions of Gentry and Waters [GW09] (denoted (a), (b), (a)-ROM); the construction by Ren et al. [RNZ14]; and the construction by Attrapadung and Libert [AL10]; all other schemes listed in the tables are secure only in the weaker selective identity model.

3. The scheme in [RNZ14] has a security degradation of $O(n^\ell)$ (where $n$ is the length of identities) thus making the reduction meaningful only for very small values of $\ell$.

4. The schemes in [GW09] achieve adaptive security but, are based on non-static and (also non-standard) assumptions. Apart from $\mathcal{IBBE}_1$, $\mathcal{IBBE}_2$, the other scheme which achieves adaptive security based on standard assumptions is [AL10].

5. The constructions in [GW09] have better ciphertext sizes whereas our constructions have better public parameter sizes. The trade-off is the use of a non-static assumption, i.e., the hardness assumption is parameterised by $m$. Another trade-off is provided by the constructions in [AL10] which provide ciphertexts consisting of a constant number of group elements while the number of group element in the decryption keys is $O(m)$.

6. Observe that our schemes require more scalar multiplications in $\mathbb{G}_1$ during encryption compared to the schemes of Gentry and Waters [GW09]. On the other hand, the number of multiplications in $\mathbb{Z}_p$ for the schemes in [GW09] is exponential in $\ell$ and becomes inefficient even for moderate values of $\ell$.

# 2 Preliminaries

In this section, we define some notation, then review pairings, complexity assumptions required for the proofs, and formal definitions related to identity-based broadcast encrytion.

## 2.1 Notation

The notation $x_1, \ldots, x_k \xleftarrow{\mathrm{R}} \mathcal{X}$ indicates that elements $x_1, \ldots, x_k$ are sampled independently from the set $\mathcal{X}$ according to some distribution R. We use U to denote the uniform distribution and so in particular, the notation $x_1, \ldots, x_k \xleftarrow{\mathrm{U}} \mathcal{X}$ denotes the independent and uniform random choice of $x_1, \ldots x_k$ from $\mathcal{X}$.

For a (probabilistic) algorithm $\mathcal{A}$, $y \xleftarrow{\mathrm{R}} \mathcal{A}(x)$ means that $y$ is chosen according to the output distribution of $\mathcal{A}$ on input $x$. A probabilistic algorithm $\mathcal{A}$ requires internal random coins for its execution. The notation $\mathcal{A}(x; r)$ denotes that $\mathcal{A}$ is run on input $x$ with its internal random coins set to $r$.

For two integers $a < b$, the notation $[a,b]$ represents the set $\{x \in \mathbb{Z} : a \leq x \leq b\}$. If $\mathbb{G}$ is a finite cyclic group, then $\mathbb{G}^{\times}$ denotes the set of generators of $\mathbb{G}$. For $p$ a prime, considering $\mathbb{Z}_p$ as an additive cyclic group, the set $\mathbb{Z}_p^{\times}$ denotes the set of all generators of $\mathbb{Z}_p$ which is the set of all non-zero elements of $\mathbb{Z}_p$.

## 2.2 Asymmetric Pairings and Hardness Assumptions

A bilinear pairing ensemble is a 7-tuple $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ where $\mathbb{G}_1 = \langle P_1 \rangle$, $\mathbb{G}_2 = \langle P_2 \rangle$ are written additively and $\mathbb{G}_T$ is a multiplicatively written group, all having the same order $p$ and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a map with the following properties.

1. *Bilinear:* For $P_1, Q_1 \in \mathbb{G}_1$ and $P_2, Q_2 \in \mathbb{G}_2$, the following holds:
   $e(P_1, P_2 + Q_2) = e(P_1, P_2)e(P_1, Q_2)$ and $e(P_1 + Q_1, P_2) = e(P_1, P_2)e(Q_1, P_2)$.

2. *Non-degenerate:* If $e(P_1, P_2) = 1_T$, the identity element of $\mathbb{G}_T$, then either $P_1$ is the identity of $\mathbb{G}_1$ or $P_2$ is the identity of $\mathbb{G}_2$.

3. *Efficiently computable:* The function $e$ should be efficiently computable.

Three main types of pairings have been identified in the literature [SV07, GPS08].

**Type-1** In this type, the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are the same.

**Type-2** $\mathbb{G}_1 \neq \mathbb{G}_2$ and an efficiently computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is known.

**Type-3** Here, $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ are known.

The constructions we provide are based on Type-3 pairings. The computational assumptions on which the security of our constructions are based are the decision Diffie-Hellman (DDH) assumptions in groups $\mathbb{G}_1$ and $\mathbb{G}_2$, called DDH1 and DDH2 assumptions respectively. We do not directly use the DDH1 assumption. Instead, we use a variant which we call the DDH1$^*$ assumption. Below, we describe these assumptions and show that if the DDH1 assumption holds then so does the DDH1$^*$ assumption also holds.

Let $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ be a Type-3 bilinear pairing ensemble.

**DDH1.** Let $\mathscr{A}$ be a probabilistic algorithm which takes as input $(\mathcal{G}, Q_1, R_1, S_1)$ and returns a bit; where $Q_1, R_1$ and $S_1$ are elements of $\mathbb{G}_1$. Let $a, b, c \xleftarrow{\text{U}} \mathbb{Z}_p$. The advantage of $\mathscr{A}$ in solving the problem DDH1 is defined to be

$$\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{G}, aP_1, bP_1, abP_1) = 1] - \Pr[\mathscr{A}(\mathcal{G}, aP_1, bP_1, (ab + c)P_1) = 1]|.$$

The probabilities are over uniform random choices of $a, b, c$ and the internal random bits of $\mathscr{A}$. The $(\varepsilon, t)$-DDH1 assumption is that, for any $t$-time algorithm $\mathscr{A}$, $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A}) \leq \varepsilon$.

**DDH2.** Let $\mathscr{A}$ be a probabilistic algorithm which takes as input $(\mathcal{G}, Q_2, R_2, S_2)$ and returns a bit; where $Q_2, R_2$ and $S_2$ are elements of $\mathbb{G}_2$. Let $a, b, c \xleftarrow{\text{U}} \mathbb{Z}_p$. The advantage of $\mathscr{A}$ in solving the problem DDH2 is defined to be

$$\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH2}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{G}, aP_2, bP_2, abP_2) = 1] - \Pr[\mathscr{A}(\mathcal{G}, aP_2, bP_2, (ab + c)P_2) = 1]|.$$

The probabilities are over uniform random choices of $a, b, c$ and the internal random bits of $\mathscr{A}$. The $(\varepsilon, t)$-DDH2 assumption is that, for any $t$-time algorithm $\mathscr{A}$, $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH2}}(\mathscr{A}) \leq \varepsilon$.

In our security reduction, we will require a slightly different form for the DDH1 problem. This is defined below.

**DDH1$^*$.** Let $\mathscr{A}$ be a probabilistic algorithm that takes as input a tuple $(\mathcal{G}, Q_1, S_1, R_1)$ and outputs a bit; where $Q_1, S_1$ and $R_1$ are elements of $\mathbb{G}_1$. Let $a \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p^\times$ and $s, \mu \overset{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p$. The advantage of $\mathscr{A}$ in solving the DDH1$^*$ problem is defined to be

$$\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}^*}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{G}, aP_1, asP_1, sP_1) = 1] - \Pr[\mathscr{A}(\mathcal{G}, aP_1, asP_1, (s + \mu)P_1) = 1]|.$$

The probabilities are over uniform random choices of $a$ in $Z_p^\times$ and $s, \mu$ in $Z_p$ and the internal random bits of $\mathscr{A}$. The $(\varepsilon, t)$-DDH1$^*$ assumption holds in $\mathcal{G}$ if for any adversary $\mathscr{A}$ running in time at most $t$, $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}^*}(\mathscr{A}) \leq \varepsilon$.

**Note.** In the above, the bound $\varepsilon$ on the advantage depends on the size of $p$ and the sizes of the groups $\mathbb{G}_1$ and $\mathbb{G}_2$. In practical situations, it will be required to set up the pairing ensemble $\mathcal{G}$ based on the target security level of the IBBE scheme.

**Proposition 2.1.** *1. If the $(\varepsilon, t)$-*DDH1 *assumption holds in $\mathcal{G}$ and $\varepsilon \leq (p-1)/p$, then the $(\varepsilon, t)$-*DDH1$^*$ *assumption also holds in $\mathcal{G}$.*

*2. If the $(\varepsilon, t)$-*DDH1$^*$ *assumption holds in $\mathcal{G}$, then the $(\varepsilon, t)$-*DDH1 *assumption holds in $\mathcal{G}$.*

A detailed proof of the proposition is provided in Appendix A. The implication from DDH1 assumption to DDH1$^*$ assumption (i.e., the first part of the equivalence) would be sufficient for our security proofs. For the sake of completeness, we also include the other way implication and show equivalence between the two assumptions.

## 2.3 Identity-Based Broadcast Encryption (IBBE)

Identity-based broadcast encryption (IBBE) is usually defined following the hybrid encryption (KEM-DEM) paradigm. The IBBE key encapsulation mechanism (KEM) produces a session key along with a header. This session key is used to encrypt the message via the data encapsulation mechanism (DEM). The DEM could be instantiated, for instance, to a (IND-CPA secure) symmetric key encryption scheme. The security of the IBBE would then rely on the security of the KEM. Our main interest is designing secure IBBE-KEMs. The details of the DEM are omitted and also not considered in the security proofs. Furthermore, for the sake of simplicity, we use the term IBBE in place of IBBE-KEM.

**Definition 2.1** (IBBE)**.** An IBBE scheme is defined by four probabilistic algorithms – $\mathsf{Setup}$, $\mathsf{Encap}$, $\mathsf{KeyGen}$ and $\mathsf{Decap}$. The identity space is denoted $\mathscr{I}$ and the key space for the DEM is denoted by $\mathscr{K}$.

$\mathsf{Setup}(\kappa, m)$ Takes as input a security parameter $\kappa$, the maximum number $m$ of identities in a privileged recipient group and generates the public parameters $\mathcal{PP}$ and the master secret $\mathcal{MSK}$. The algorithm also defines the identity space $\mathscr{I}$ and key space $\mathscr{K}$ for the DEM.

$\mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id})$ Input is an identity $\mathsf{id}$ and master secret $\mathcal{MSK}$; output is a secret key $\mathcal{SK}_{\mathsf{id}}$ for $\mathsf{id}$.

$\mathsf{Encap}(\mathcal{PP}, S \subseteq \mathscr{I})$ Takes as input a set of identities $S$ that are the intended recipients of the message. If $|S| \leq m$, the algorithm outputs a pair $(\mathsf{Hdr}, K)$ where $\mathsf{Hdr}$ is the header and $K \in \mathscr{K}$ is the session key.

$\mathsf{Decap}(\mathcal{PP}, S, \mathsf{id}, \mathcal{SK}_{\mathsf{id}}, \mathsf{Hdr})$ Inputs the public parameters, a set $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\}$, an identity $\mathsf{id}$, a secret key $\mathcal{SK}_{\mathsf{id}}$ corresponding to $\mathsf{id}$, a header $\mathsf{Hdr}$ and outputs the session key $K$ if $\mathsf{id} \in S$.

The message to be broadcast is encrypted using a DEM $\mathcal{S}ym = (\mathcal{S}ym.\mathsf{Encrypt}, \mathcal{S}ym.\mathsf{Decrypt})$ with key space $\mathscr{K}$. Let $\mathcal{C} \overset{\mathrm{R}}{\longleftarrow} \mathcal{S}ym.\mathsf{Encrypt}(K, M)$ where $M$ is the message to be broadcast and $K$ is the session key returned by $\mathsf{Encap}$ algorithm. The broadcast consists of the triple $(S, \mathsf{Hdr}, \mathcal{C})$. The full header is given by $(S, \mathsf{Hdr})$. During decryption, the key $K$ output by the $\mathsf{Decap}$ algorithm is used to decrypt $\mathcal{C}$ to obtain the message $M$ as $M = \mathcal{S}ym.\mathsf{Decrypt}(K, \mathcal{C})$.

**Correctness.** The IBBE scheme satisfies the correctness condition if for all sets $S \subseteq \mathscr{I}$ with $|S| \leq m$, for all $\mathsf{id}_i \in S$, if $(\mathcal{PP}, \mathcal{MSK}) \overset{\mathrm{R}}{\longleftarrow} \mathsf{Setup}(\kappa, m)$, $\mathcal{SK}_{\mathsf{id}_i} \overset{\mathrm{R}}{\longleftarrow} \mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id}_i)$, $(\mathsf{Hdr}, K) \overset{\mathrm{R}}{\longleftarrow} \mathsf{Encap}(\mathcal{PP}, S)$, then $\Pr[K = \mathsf{Decap}(\mathcal{PP}, S, \mathsf{id}_i, \mathcal{SK}_{\mathsf{id}_i}, \mathsf{Hdr})] = 1$.

**Notes:**

1. We have provided $m$ as an input to the setup algorithm. This is because of the fact that in the schemes that we describe later, the public parameters and the master secret key will depend on $m$. For a general IBBE scheme, it is not necessarily required for the setup algorithm to take $m$ as an input.

2. The setup algorithm also takes as input a security parameter $\kappa$. This indicates the security level of the scheme that is to be set up which in turn will determine the sizes of the various components of the pairing ensemble $\mathcal{G}$.

**Definition 2.2** (IBBE Security)**.** Adaptive security against chosen plaintext attacks in identity-based broadcast encryption systems is defined via the following game $\mathsf{ind\text{-}cpa}$ between an adversary $\mathscr{A}$ and a challenger.
**Setup:** The challenger runs the $\mathsf{Setup}$ algorithm of the IBBE and gives the public parameters to $\mathscr{A}$.
**Key Extraction Phase 1:** $\mathscr{A}$ makes a number of key extraction queries adaptively. For a query on an identity vector $\mathsf{id}$, the challenger responds with a key $\mathcal{SK}_{\mathsf{id}}$.
**Challenge:** $\mathscr{A}$ provides a challenge set $\widehat{S}$ with the restriction that if $\mathsf{id}$ is queried in the key extraction phase 1, then $\mathsf{id} \notin \widehat{S}$. The challenger computes $(\widehat{\mathsf{Hdr}}, K_0) \xleftarrow{\text{R}} \mathsf{Encap}(\mathcal{PP}, \widehat{S})$ and chooses $K_1 \xleftarrow{\text{U}} \mathscr{K}$. It then chooses a bit $\beta$ uniformly at random from $\{0,1\}$ and returns $(\widehat{\mathsf{Hdr}}, K_\beta)$ to $\mathscr{A}$.
**Key Extraction Phase 2:** $\mathscr{A}$ makes more key extraction queries with the restriction that it cannot query a key for any identity in $\widehat{S}$.
**Guess:** $\mathscr{A}$ outputs a bit $\beta'$.

If $\beta = \beta'$, then $\mathscr{A}$ wins the game. The advantage of $\mathscr{A}$ of the IBBE scheme in winning the $\mathsf{ind\text{-}cpa}$ game is given by

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\text{IBBE}}(\mathscr{A}) = \left| \Pr[\beta = \beta'] - \frac{1}{2} \right|.$$

The IBBE scheme is said to be $(\varepsilon, t, q)$-$\mathsf{IND\text{-}ID\text{-}CPA}$ secure if every $t$-time adversary making at most $q$ key extraction queries has $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\text{IBBE}}(\mathscr{A}) \leq \varepsilon$.

# 3 IBBE – A First Construction

Our IBBE constructions are based on a variant of the Jutla-Roy IBE [JR13] defined in [RS13] referred to as JR-IBE-D described in the following section.

## 3.1 Variant of Jutla-Roy IBE

In the dual system proof strategy, there are two kinds of decryption keys and ciphertexts called normal and semi-functional. The normal keys and ciphertexts are those which are defined in the actual scheme, while the semi-functional keys and ciphertexts are defined as part of the proof and will not be used in the actual scheme. More details on how the semi-functional keys and ciphertexts are used in the proof are given below.

For the description of the JR-IBE-D, we use a compact notation to denote normal and semi-functional ciphertexts and keys. The group elements shown in curly brackets { } are the semi-functional components. To get the scheme itself, these components should be ignored.
**Parameters:** Choose $P_1 \xleftarrow{\text{U}} \mathbb{G}_1^\times$, $P_2 \xleftarrow{\text{U}} \mathbb{G}_2^\times$, $\alpha_1, \alpha_2, \Delta_1, \Delta_2, \Delta_3, c, d, f \xleftarrow{\text{U}} \mathbb{Z}_p$, $b \xleftarrow{\text{U}} \mathbb{Z}_p^\times$, and set $U_1 = (-\Delta_1 b + d)P_1$, $V_1 = (-\Delta_2 b + f)P_1$, $W_1 = (-\Delta_3 b + c)P_1$, $g_T = e(P_1, P_2)^{\alpha_1 + b\alpha_2}$. The parameters are given by
$\mathcal{PP} : (P_1, bP_1, U_1, V_1, W_1, g_T)$
$\mathcal{MSK} : (P_2, cP_2, \alpha_1, \alpha_2, \Delta_1, \Delta_2, \Delta_3, d, f)$
**Encryption:**
$\mathsf{tag}, s \xleftarrow{\text{U}} \mathbb{Z}_p$, $\{\mu \xleftarrow{\text{U}} \mathbb{Z}_p\}$
$C_0 = m \cdot (g_T)^s \{\times e(P_1, P_2)^{u\mu}\}$,
$C_1 = sP_1\{+\mu P_1\}$, $C_2 = sbP_1$, $C_3 = s(U_1 + \mathsf{id}V_1 + \mathsf{tag}W_1)\{+\mu(d + \mathsf{id} \cdot f + \mathsf{tag} \cdot c)P_1\}$.
**Key Generation:**

$r \xleftarrow{\text{U}} \mathbb{Z}_p$, $\{\gamma, \pi \xleftarrow{\text{U}} \mathbb{Z}_p\}$
$K_1 = rP_2$, $K_2 = rcP_2\{+\gamma P_2\}$, $K_3 = (\alpha_1 + r(d + \mathsf{id}f))\, P_2\{+\gamma\pi P_2\}$,
$K_4 = -r\Delta_3 P_2\{-\frac{\gamma}{b} P_2\}$, $K_5 = (\alpha_2 - r(\Delta_1 + \mathsf{id}\Delta_2))\, P_2\{-\frac{\gamma\pi}{b} P_2\}$.

**Decryption:** Given ciphertext $\mathcal{C} = (C_0, C_1, C_2, C_3, \mathsf{tag})$ and key $\mathcal{SK}_{\mathsf{id}} = (K_1, \ldots, K_5)$, the message is recovered as follows:

$$m = C_0 \cdot \frac{e(C_3, K_1)}{e(C_1, K_2 + \mathsf{tag} \cdot K_3)e(C_2, K_4 + \mathsf{tag} \cdot K_5)} .$$

## 3.2 Overview of the IBBE Construction

We start by providing a brief overview of our first IBBE construction – $\mathcal{IBBE}_1$. The starting point is JR-IBE-D that achieves adaptive-identity security from the DDH assumptions in $\mathbb{G}_1$ and $\mathbb{G}_2$. Let $N_1, N_2, N_T$ and $N_p$ denote the sizes of representation of elements in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and $\mathbb{Z}_p$ respectively. A ciphertext in JR-IBE-D consists of the three elements $C_1, C_2$ and $C_3$ from $\mathbb{G}_1$; the element $C_0$ from $\mathbb{G}_T$; and the element $\mathsf{tag}$ from $\mathbb{Z}_p$. The size of one ciphertext is $N_T + 3N_1 + N_p$.

Now consider the setting of identity-based broadcast encryption. Suppose that $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\} \subseteq \mathscr{I}$ is a set of identities corresponding to the intended recipients of a message. A natural way to extend the IBE scheme to the broadcast setting is as follows. The user keys will be the usual IBE decryption keys and the public parameters will also remain the same. Components $C_1, C_2$ would still remain the same since they are independent of the identity. The mask $(g_T)^s$ used to encrypt the message in $C_0$ will now play the role of the session key i.e., $K = (g_T)^s$. Introduce separate identity-hashes for each identity but randomised with the same scalar. In particular, $C_3$ is replaced by $C_{3,i} = s(U_1 + \mathsf{id}_i V_1 + \mathsf{tag}_i W_1)$, $i \in [1, \ell]$.

We would like to emphasise that having separate hashes for each identity requires the use of separate tags for the different hashes. Otherwise, one can get hold of $sV_1$ by just taking the difference between $C_{3,i}$ and $C_{3,j}$ for some $i \neq j$. With $sV_1$, an attacker can construct a header for $S' = S \cup \{\mathsf{id}\}$ for any $\mathsf{id}$ of its choice. This header when decapsulated using a secret key for $\mathsf{id}$, results in the same session key that the header for $S$ encapsulates. So, not having separate tags makes the scheme insecure.

For the scheme with separate tags as described above the header size will be $(2+\ell)N_1 + \ell N_p$. This is better than performing separate IBE encryptions for each identity resulting in header size of $\ell(N_T + 3N_1 + N_p)$. However, the scheme as described does not seem to admit a security proof. Defining $C_{3,i}$ as above leads to problems during simulation within the dual system framework. To see why the above method fails, we take a look at the dual system proof of JR-IBE-D.

**The structure of dual-system proof:** In a dual system proof for IBE, two types of ciphertexts and keys are defined – one is normal (as generated in the scheme) and the other is semi-functional (defined using some secret information possibly available only in the master secret). The proof is organised as a hybrid over a sequence of games where the challenge ciphertext and the secret keys returned as responses to key extraction queries are changed to semi-functional form. Once this is done, a final game is defined where the message encrypted by the challenge ciphertext is switched to random. This is mainly to argue about indistinguishability of ciphertexts. The security guarantee is obtained by showing that any two successive games are indistinguishable based on the hardness of some problems (DDH1, DDH2 in case of JR-IBE-D).

To this end, an important step is to show that a normal key is computationally indistinguishable from a semi-functional key. When the attacker requests a key for an identity $\mathsf{id}$, a DDH2 instance is embedded in the key $\mathcal{SK}_{\mathsf{id}}$ in such a way that the power of the attacker in determining whether $\mathcal{SK}_{\mathsf{id}}$ is normal or semi-functional can be used to solve the particular instance. At the same time, it is required to create a valid semi-functional ciphertext for the challenge identity $\widehat{\mathsf{id}}$. One must also ensure any semi-functional ciphertext that is created for $\mathsf{id}$ cannot provide any extra advantage in solving the problem instance. All this is achieved by embedding a degree one polynomial $f(x) = Ax + B$ in both the $\widehat{\mathsf{tag}}$ in the ciphertext for $\widehat{\mathsf{id}}$ and the scalar $\pi$ in the semi-functional components of $\mathcal{SK}_{\mathsf{id}}$. Moreover, $A$ and $B$ are programmed into the public parameters in such a way that they are information theoretically hidden from an attacker's viewpoint. Specifically, they are embedded in parameters $V_1$ and $U_1$ in the $\mathcal{PP}$.

First of all, a degree one polynomial in random variables $A, B$ provides pairwise independence when evaluated at two different points ($A, B$ are uniformly and independently distributed). This ensures correct distribution of $\pi = f(\mathsf{id})$ and $\mathsf{tag} = f(\widehat{\mathsf{id}})$. Secondly, the only way of creating a semi-functional ciphertext for an identity $\mathsf{id}'$ is by setting $\mathsf{tag}' = f(\mathsf{id}')$ implying that any attempt to create a semi-functional ciphertext for $\mathsf{id}$ will set $\mathsf{tag} = \pi$. As a result, decryption is successful and no information is gained regarding the semi-functionality of $\mathcal{SK}_{\mathsf{id}}$.

**Independence issue for IBBE scheme:** In the extension to the broadcast setting discussed above, we need to argue about the independence of $\widehat{\ell}$ tags $\mathsf{tag}_1, \ldots, \mathsf{tag}_{\widehat{\ell}}$ in the challenge header for $\widehat{S} = \{\widehat{\mathsf{id}}_1, \ldots, \widehat{\mathsf{id}}_{\widehat{\ell}}\}$, plus the scalar $\pi$ in the secret key for some $\mathsf{id} \notin \widehat{S}$. Also we need to argue about the joint distribution of all the tags in a single step since they all share the same randomiser. A degree one polynomial does not provide sufficient amount of randomness to do so. This is exactly where the dual system argument fails.

To overcome this problem, we introduce the restriction that the maximum size of a privileged users' should be at most $m$. Then we replace the JR-IBE-D identity hash by a degree-$m$ polynomial hash in the identity. Such a polynomial provides $(m+1)$-wise independence. Since one needs to argue about the independence of at most $m$ tags and one $\pi$, this hash will suffice for a dual system proof.

The coefficients of the polynomial are determined by the public parameters. So instead of $U_1, V_1$, $\mathcal{PP}$ will now contain elements $U_{1,j}$ for $j = 0, \ldots, m$. Define component $C_{3,i}$ as $C_{3,i} = s(\sum_{j=0}^{m} (\mathsf{id}_i)^j U_{1,j} + \mathsf{tag}_i W_1)$ for $\mathsf{id}_i \in S$. Also, as in JR-IBE-D, $U_{1,j}$'s and $W_1$ are created using linear combinations of certain scalars in the master secret i.e., $U_{1,j} = (e_j + b\Delta_j)P_1$ for $j = 0, \ldots, m$ and $W_1 = (c + b\Delta)P_1$. So the secret key for an identity $\mathsf{id}$ will now consist of the two sub-hashes $\sum_{j=0}^{m} (\mathsf{id})^j e_j$ and $\sum_{j=0}^{m} (\mathsf{id})^j \Delta_j$. These sub-hashes are combined using $b$ in $C_2$ during decryption to cancel out the hash in $C_{3,i}$ if $\mathsf{id} = \mathsf{id}_i$.

The technique of using polynomials to hash identities has been used earlier by Chatterjee and Sarkar in [CS06a] in the context of IBBE. However, they only obtain weaker security against selective-identity attacks.

## 3.3 Construction of $\mathcal{IBBE}_1$

Our first IBBE construction is $\mathcal{IBBE}_1 = (\mathcal{IBBE}_1.\mathsf{Setup}, \mathcal{IBBE}_1.\mathsf{Encrypt}, \mathcal{IBBE}_1.\mathsf{KeyGen}, \mathcal{IBBE}_1.\mathsf{Decrypt})$ whose description is given in Figure 1.

**Correctness:** Let $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\} \subseteq \mathscr{I}$ with $\ell \leq m$. Let $(\mathsf{Hdr}, K) \longleftarrow \mathcal{IBBE}_1.\mathsf{Encap}(\mathcal{PP}, S; s)$ where $\mathsf{Hdr} = (C_1, C_2, (C_{3,i}, \mathsf{tag}_i)_{i=1}^{\ell})$ and let $\mathcal{SK}_{\mathsf{id}_i} \xleftarrow{\text{R}} \mathcal{IBBE}_1.\mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id}_i; r)$ for some $\mathsf{id}_i \in S$.

$$\begin{aligned}
&\frac{e(C_1, \mathsf{tag}_i D_2 + D_3) e(C_2, \mathsf{tag}_i D_4 + D_5)}{e(C_{3,i}, D_1)} \\
&= \frac{e(sP_1, \mathsf{tag}_i \cdot rcP_2 + (\alpha_1 + r(\sum_{j=0}^{m} (\mathsf{id}_i)^j e_j))P_2) \cdot e(sbP_1, \mathsf{tag}_i r\Delta P_2 + (\alpha_2 + r(\sum_{j=0}^{m} (\mathsf{id}_i)^j \Delta_j))P_2)}{e(s(\sum_{j=0}^{m} (\mathsf{id}_i)^j U_{1,j} + \mathsf{tag}_i W_1), rP_2)} \\
&= \frac{e(sP_1, \alpha_1 P_2) \cdot e(sP_1, P_2)^{\mathsf{tag}_i rc + r(\sum_{j=0}^{m} (\mathsf{id}_i)^j e_j)} \cdot e(sP_1, b\alpha_2 P_2) e(sP_1, P_2)^{\mathsf{tag}_i \cdot r\Delta b + r(\sum_{j=0}^{m} (\mathsf{id}_i)^j \Delta_j b)}}{e((\sum_{j=0}^{m} (\mathsf{id}_i)^j \Delta_j b + \mathsf{tag}_i \Delta b + \sum_{j=0}^{m} (\mathsf{id}_i)^j e_j + \mathsf{tag}_i c)P_1, P_2)^{rs}} \\
&= \frac{e(P_1, (\alpha_1 + b\alpha_2)P_2)^s \cdot e((\sum_{j=0}^{m} (\mathsf{id}_i)^j \Delta_j b + \mathsf{tag}_i \Delta b + \sum_{j=0}^{m} (\mathsf{id}_i)^j e_j + \mathsf{tag}_i c)P_1, P_2)^{rs}}{e((\sum_{j=0}^{m} (\mathsf{id}_i)^j \Delta_j b + \mathsf{tag}_i \Delta b + \sum_{j=0}^{m} (\mathsf{id}_i)^j e_j + \mathsf{tag}_i c)P_1, P_2)^{rs}} \\
&= g_T^s.
\end{aligned}$$

**Header size and user storage:** The header consists of $(2 + \ell)$ elements of $\mathbb{G}_1$, $\ell$ elements of $\mathbb{Z}_p$ and one element of $\mathbb{G}_T$. Using the previous notation, the size of the header is $(2 + \ell)N_1 + \ell N_p + N_T$. The number of keys to be stored by each user consists of 5 elements of $\mathbb{G}_2$.

Figure 1: Construction of $\mathcal{IBBE}_1$.

```
$\mathcal{IBBE}_1$.Setup$(\kappa, m)$

1. Generate $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, F_1, F_2)$ based on $\kappa$.
2. Set $\mathscr{I} = \mathbb{Z}_p$ and $\mathscr{K} = \mathbb{G}_T$.
3. Set $P_1 \xleftarrow{\text{U}} \mathbb{G}_1^{\times}$, $P_2 \xleftarrow{\text{U}} \mathbb{G}_2^{\times}$.
4. Choose $\alpha_1, \alpha_2, c, \Delta, (e_j, \Delta_j)_{j=0}^m \xleftarrow{\text{U}} \mathbb{Z}_p$.
5. Choose $b \xleftarrow{\text{U}} \mathbb{Z}_p^{\times}$.
6. For $j = 0, \ldots, m$
      set $U_{1,j} = (\Delta_j b + e_j)P_1$.
7. Set $W_1 = (\Delta b + c)P_1$.
8. Set $g_T = e(P_1, P_2)^{\alpha_1 + b\alpha_2}$.
Define $\mathcal{PP} : (P_1, bP_1, (U_{1,j})_{j=0}^m, W_1, g_T)$.
Define $\mathcal{MSK} : (P_2, cP_2, \alpha_1, \alpha_2, \Delta, (e_j, \Delta_j)_{j=0}^m)$.
```

```
$\mathcal{IBBE}_1$.KeyGen$(\mathcal{MSK}, \mathsf{id})$

1. Choose $r \xleftarrow{\text{U}} \mathbb{Z}_p$.
2. Set $D_1 = rP_2$.
3. Set $D_2 = rcP_2$.
4. Set $D_3 = \Big(\alpha_1 + r(\sum_{j=0}^m (\mathsf{id})^j e_j)\Big) P_2$.
5. Set $D_4 = r\Delta P_2$.
6. Set $D_5 = \Big(\alpha_2 + r(\sum_{j=0}^m (\mathsf{id})^j \Delta_j)\Big) P_2$.
Return $\mathcal{SK}_{\mathsf{id}} = (D_1, D_2, D_3, D_4, D_5)$.
```

```
$\mathcal{IBBE}_1$.Encap$(\mathcal{PP}, S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\})$.

1. If $\ell \leq m$, pick $s, (\mathsf{tag}_i)_{i=1}^{\ell} \xleftarrow{\text{U}} \mathbb{Z}_p$.
2. Compute the session key as $K = g_T^s$.
3. Set $C_1 = sP_1$, $C_2 = sbP_1$.
4. For $i = 1, \ldots, \ell$
      set $C_{3,i} = s(\sum_{j=0}^m (\mathsf{id}_i)^j U_{1,j} + \mathsf{tag}_i W_1)$.
5. $\mathsf{Hdr} = (C_1, C_2, (C_{3,i}, \mathsf{tag}_i)_{i=1}^{\ell})$.
Return $(\mathsf{Hdr}, K)$.
```

```
$\mathcal{IBBE}_1$.Decap$(\mathcal{PP}, S, \mathsf{id}, \mathcal{SK}_{\mathsf{id}}, \mathsf{Hdr})$

1. Let $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\}$.
2. If $\mathsf{id} \in S$, find $i \in [1, \ell]$ such that $\mathsf{id} = \mathsf{id}_i$.
3.   Compute
        $A = e(C_1, \mathsf{tag}_i D_2 + D_3) e(C_2, \mathsf{tag}_i D_4 + D_5)$.
        $B = e(C_{3,i}, D_1)$.
        $K = A/B$.
4. Else $K = \perp$.
Return $K$.
```

**Use of random oracles.** Let $H : \{0,1\}^\kappa \times [1,m] \to \mathbb{Z}_p$ be a hash function that takes a seed (say $z$) of length $\kappa$, an index $i \in [1,m]$ as input and produces a value in $\mathbb{Z}_p$ as output. We show how to apply a technique from [GW09] to reduce the header size. If $H$ is modelled as a random oracle, then for distinct inputs, the outputs will be independent and uniformly distributed in $\mathbb{Z}_p$. Such an $H$ can be used to reduce the header size in the following manner. In the $\mathcal{IBBE}_1$ header, the tags are replaced by a uniform random $\kappa$-bit quantity $z$. The actual tags are generated by evaluating $H$ on inputs $(z,i)$ for each $i \in [1,\ell]$ where $|S| = \ell$. The size of the resulting header will be $N_T + (2+\ell)N_1 + \kappa$. In practical terms, the efficiency gain over $\mathcal{IBBE}_1$ is quite significant. The modified scheme which we call $\mathcal{IBBE}_1^{\mathrm{RO}}$ (RO denotes random oracle), can be shown to be secure via a reduction from an adversary breaking its security to an adversary against scheme $\mathcal{IBBE}_1$. Essentially, the tags that the adversary against $\mathcal{IBBE}_1$ obtains as part of the challenge header are returned as answers to the random oracle queries that the adversary against $\mathcal{IBBE}_1^{\mathrm{RO}}$ makes. Note that the use of random oracles is “minimal”. It may be possible to use ROs more effectively to further reduce the header size.

**Getting rid of tags?** It would be nice to be able to completely get rid of the tags. These tags play a crucial role in the dual system proof. Lewko and Waters [LW10] proposed a different type of dual system encryption where the role of the tags is shifted to some scalars in the semi-functional components (similar to the scalar $\pi$ in a $\mathcal{IBBE}_1$ secret key). However, one must also ensure that a semi-functional component can be decrypted by a normal key which in turn requires that these scalars in the semi-functional components cancel out during decryption. This can be done with multiple copies of the identity hash (as in [LW10]) in the ciphertext. In the context of broadcast encryption, having multiple copies of the identity hash in the ciphertext increases the header size. So, it does not seem likely that the technique of [LW10] will help reduce the header size any further.

**Restriction on the size of the identity set:** In the encapsulation algorithm we have assumed that the number of identities $\ell$ to which the message is to be encrypted is at most $m$, the parameter of the IBBE scheme. If it turns out that $\ell > m$, then the set of identities will be divided into $\lceil \ell/m \rceil$ groups and the encapsulation algorithm will be applied separately to each group. The resulting header size will be $\lceil \ell/m \rceil((m+2)N_1 + mN_p + N_T)$. Since this is quite routine, we will simply analyse the scheme under the assumption that $\ell \le m$.

### 3.4 Security of $\mathcal{IBBE}_1$

The scheme $\mathcal{IBBE}_1$ is proved secure in the sense of IND-ID-CPA (Section 2.3, Definition 2.2) via the dual system technique. The following theorem formally states the security guarantee we prove for the scheme $\mathcal{IBBE}_1$.

**Theorem 3.1.** *If $(\varepsilon_{\text{DDH1}}, t_1)$-DDH1 and $(\varepsilon_{\text{DDH2}}, t_2)$-DDH2 assumptions hold in $\mathcal{G}$, then $\mathcal{IBBE}_1$ is $(\varepsilon, t, q)$-IND-ID-CPA-secure where $\varepsilon \le \varepsilon_{\text{DDH1}} + q \cdot \varepsilon_{\text{DDH2}} + 1/p$, $t_1 = t + O(m^2\rho)$ and $t_2 = t + O(m^2\rho)$. Here $\rho$ is the maximum time required for one scalar multiplication in $\mathbb{G}_1$ or $\mathbb{G}_2$.*

*Proof.* By Proposition 2.1, since the $(\varepsilon_{\text{DDH1}}, t_1)$-DDH1 assumption holds in $\mathcal{G}$, the $(\varepsilon_{\text{DDH1}}, t_1)$-DDH1* assumption also holds in $\mathcal{G}$. In our proof, we will use the later assumption.

We start by appropriately defining semi-functional headers and user keys for $\mathcal{IBBE}_1$. Let $\mathcal{IBBE}_1$.SFEncap and $\mathcal{IBBE}_1$.SFKeyGen be algorithms that generate semi-functional headers and user keys (respectively) described as follows.

$\mathcal{IBBE}_1$.SFEncap$(\mathcal{PP}, \mathcal{MSK}, S, (\mathsf{Hdr}, K))$: Takes as input a header-session key pair $(\mathsf{Hdr}, K)$ created by $\mathcal{IBBE}_1$.Encap algorithm on a set $S$ and modifies it to obtain semi-functional header and session key. Let $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\}$ and $\mathsf{Hdr} = (C_1, C_2, (C_{3,i}, \mathsf{tag}_i)_{i=1}^{\ell})$. Pick $\mu \xleftarrow{\text{U}} \mathbb{Z}_p$ and modify $K$ and the components of Hdr as follows.

$$K \leftarrow K \cdot e(P_1, P_2)^{\alpha_1 \mu}, \quad C_1 \leftarrow C_1 + \mu P_1, \quad C_2 \leftarrow C_2,$$

$$C_{3,i} \leftarrow C_{3,i} + \mu \left( \sum_{j=0}^{m} (\mathsf{id}_i)^j e_j + \mathsf{tag}_i \cdot c \right) P_1 \text{ for } i = 1, \ldots, \ell.$$

Return the modified session key $K$ along with the header $\mathsf{Hdr} = (C_1, C_2, (C_{3,i}, \mathsf{tag}_i)_{i=1}^{\ell})$.
$\mathcal{IBBE}_1$.$\mathsf{SFKeyGen}(\mathcal{MSK}, \mathcal{SK}_{\mathsf{id}})$: This algorithm takes in a normal secret key $\mathcal{SK}_{\mathsf{id}} = (D_1, \ldots, D_5)$ for identity $\mathsf{id}$ and generates a semi-functional key as follows.

$\gamma, \pi \xleftarrow{\text{U}} \mathbb{Z}_p$,

$$D_1 \leftarrow D_1, \quad D_2 \leftarrow D_2 + \gamma P_2, \quad D_3 \leftarrow D_3 + \gamma\pi P_2,$$
$$D_4 \leftarrow D_4 - \left(\frac{\gamma}{b}\right) P_2, \quad D_5 \leftarrow D_5 - \left(\frac{\gamma\pi}{b}\right) P_2.$$

The resulting key $\mathcal{SK}_{\mathsf{id}} = (D_1, \ldots, D_5)$ is returned.

We need to show that all the semi-functionality properties are satisfied. Let $(\mathsf{Hdr} = (C_1, C_2, (C_{3,i}, \mathsf{tag}_i)_{i=1}^{\ell}), K)$ be a header-key pair for the set $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\}$ and let $\mathcal{SK}_{\mathsf{id}_i}$ be a user key for an identity $\mathsf{id}_i \in S$. Consider the following cases.

**$\mathcal{SK}_{id_i}$ is semi-functional and $(\mathsf{Hdr}, K)$ is normal:** Let $\mathcal{SK}'_{id_i}$ be a normally generated key for $\mathsf{id}_i$ and $\mathcal{SK}_{\mathsf{id}_i} \longleftarrow \mathcal{IBBE}_1.\mathsf{SFKeyGen}(\mathcal{MSK}, \mathcal{SK}'_{\mathsf{id}_i}; \gamma, \pi)$. The requirement is that when $\mathsf{Hdr}$ is decapsulated with $\mathcal{SK}_{\mathsf{id}}$, the result is $K$. The following calculation shows that this requirement is satisfied.

$$\begin{aligned} &\frac{e(C_1, \mathsf{tag}_i D_2 + D_3) e(C_2, \mathsf{tag}_i D_4 + D_5)}{e(C_{3,i}, D_1)} \\ &\quad = K \cdot e(sP_1, \mathsf{tag}_i \gamma P_2 + \gamma\pi P_2) e(sbP_1, -\mathsf{tag}_i(\gamma/b)P_2 - (\gamma\pi/b)P_2) \\ &\quad = K \cdot e(sP_1, \mathsf{tag}_i \gamma P_2 + \gamma\pi P_2) e(sP_1, -\mathsf{tag}_i \gamma P_2 - \gamma\pi P_2) \\ &\quad = K. \end{aligned}$$

The second step follows from the correctness condition i.e., a normal header when decapsulated with a normal user key gives the corresponding normal session key.

**$\mathcal{SK}_{\mathsf{id}_i}$ is normal and $(\mathsf{Hdr}, K)$ is semi-functional:** Let $(\mathsf{Hdr}', K')$ be a normally generated header-key pair and let $(\mathsf{Hdr}, K) \longleftarrow \mathcal{IBBE}_1.\mathsf{SFEncap}(\mathcal{PP}, \mathcal{MSK}, S, (\mathsf{Hdr}', K'); \mu)$. We have

$$\begin{aligned} &\frac{e(C_1, \mathsf{tag}_i D_2 + D_3) e(C_2, \mathsf{tag}_i D_4 + D_5)}{e(C_{3,i}, D_1)} \\ &\quad = K \cdot \frac{e(\mu P_1, \mathsf{tag}_i D_2 + D_3)}{e(\mu(\sum_{j=0}^{m} (\mathsf{id}_i)^j e_j + \mathsf{tag}_i \cdot c) P_1, D_1)} \\ &\quad = K \cdot \frac{e(\mu P_1, r(\sum_{j=0}^{m} (\mathsf{id}_i)^j e_j + \mathsf{tag}_i \cdot c) P_2)}{e(\mu(\sum_{j=0}^{m} (\mathsf{id}_i)^j e_j + \mathsf{tag}_i \cdot c) P_1, rP_2)} \\ &\quad = K, \end{aligned}$$

as required.

**Both $\mathcal{SK}_{\mathsf{id}_i}$ and $(\mathsf{Hdr}, K)$ are semi-functional:** Let $(\mathsf{Hdr}', K')$ be a normally generated header-key pair and $\mathcal{SK}'_{\mathsf{id}_i}$ a normal key for $\mathsf{id}_i$. Let $\mathcal{SK}_{\mathsf{id}_i} \longleftarrow \mathcal{IBBE}_1.\mathsf{SFKeyGen}(\mathcal{MSK}, \mathcal{SK}'_{\mathsf{id}_i}; \gamma, \pi)$ and $(\mathsf{Hdr}, K) \longleftarrow \mathcal{IBBE}_1.\mathsf{SFEncap}(\mathcal{PP}, \mathcal{MSK}, S, (\mathsf{Hdr}', K'); \mu)$. In this case, the key obtained by running the $\mathcal{IBBE}_1$.$\mathsf{Decap}$ algorithm is masked by a factor of $e(P_1, P_2)^{\mu\gamma(\mathsf{tag}_i + \pi)}$ as shown below.

$$\begin{aligned} &\frac{e(C_1, \mathsf{tag}_i D_2 + D_3) e(C_2, \mathsf{tag}_i D_4 + D_5)}{e(C_{3,i}, D_1)} \\ &\quad = K \cdot e(\mu P_1, \mathsf{tag}_i \gamma P_2 + \gamma\pi P_2) \\ &\quad = K \cdot e(P_1, P_2)^{\mu\gamma(\mathsf{tag}_i + \pi)}. \end{aligned}$$

In the second step we retain only pairings between semi-functional components since all other pairings involving semi-functional components get cancelled.

Note that the masking factor vanishes when $\mathsf{tag}_i = -\pi$. Then $\mathcal{SK}_{id_i}$ and the $i$-th component of $\mathsf{Hdr}$ are called *nominally semi-functional.*

Now, given that semi-functional algorithms are defined, consider a sequence of games $\mathsf{G}_{real}$, $\mathsf{G}_0$, $(\mathsf{G}_k)_{k=1}^q$, $\mathsf{G}_{final}$ between an adversary $\mathscr{A}$ and a challenger with the games defined as follows. Recall that $q$ is the number of key extraction queries made by the adversary.

- $\mathsf{G}_{real}$: the actual IBBE security game $\mathsf{ind\text{-}cpa}$ (described in Section 2.3).
- $\mathsf{G}_k$, $0 \le k \le q$: challenge header is semi-functional; $K_0$ is semi-functional; first $k$ user keys are semi-functional.
- $\mathsf{G}_{final}$: challenge header is semi-functional and the adversary's advantage in guessing the bit $\beta$ is at most $1/p$.

Let $X_\Box$ denote the event that $\mathscr{A}$ wins in $\mathsf{G}_\Box$. In Lemmas 3.1, 3.2 and 3.3, we show that

- $|\Pr[X_{real}] - \Pr[X_0]| \le \varepsilon_{\text{DDH1}^*} \le \varepsilon_{\text{DDH1}}$,
- $|\Pr[X_{k-1}] - \Pr[X_k]| \le \varepsilon_{\text{DDH2}}$,
- $\Pr[X_q] = \Pr[X_{final}]$ and $|\Pr[X_{final}] - 1/2| \le 1/p$.

Hence, the advantage of $\mathscr{A}$ in breaking the security of $\mathscr{IBBE}_1$ is thus given by

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathscr{IBBE}_1}(\mathscr{A}) &= |\Pr[X_{real}] - \frac{1}{2}| \\
&\le |\Pr[X_{real}] - \Pr[X_{final}]| + |\Pr[X_{final}] - \frac{1}{2}| \\
&\le |\Pr[X_{real}] - \Pr[X_0]| + \sum_{k=1}^{q}(|\Pr[X_{k-1}] - \Pr[X_k]|) \\
&\qquad + |\Pr[X_q] - \Pr[X_{final}]| + \frac{1}{p} \\
&\le \varepsilon_{\text{DDH1}} + q\varepsilon_{\text{DDH2}} + \frac{1}{p}.
\end{aligned}$$

□

In the sequel, $\mathscr{B}_1$ (resp. $\mathscr{B}_2$) is a DDH1*-solver (resp. DDH2-solver). We argue that $\mathscr{B}_1$, using the adversary's ability to distinguish between $\mathsf{G}_{real}$ and $\mathsf{G}_0$, can solve DDH1*. Similarly, $\mathscr{A}$'s power to distinguish between $\mathsf{G}_{k-1}$ and $\mathsf{G}_k$ for $k \in [1, q]$, can be leveraged to build a DDH2-solver $\mathscr{B}_2$.

**Lemma 3.1.** $|\Pr[X_{real}] - \Pr[X_0]| \le \varepsilon_{\text{DDH1}}$.

*Proof.* Let $(\mathcal{G}, bP_1, sbP_1, P_2, (s+\mu)P_1)$ be the instance of DDH1* that $\mathscr{B}_1$ has to solve i.e., decide whether $\mu = 0$ or $\mu \in_{\text{U}} \mathbb{Z}_p$. Note that by the definition of the DDH1* problem, $b \neq 0$. The phases of the game are simulated by $\mathscr{B}_1$ as described below.
**Setup:** Choose $\alpha_1, \alpha_2, c, \Delta, (e_j, \Delta_j)_{j=0}^m \xleftarrow{\text{U}} \mathbb{Z}_p$ and set parameters as:
$U_{1,j} = \Delta_j(bP_1) + e_j P_1$ for $j = 0, \ldots, m$, $W_1 = \Delta(bP_1) + cP_1$,
$g_T = e(P_1, P_2)^{\alpha_1} e(bP_1, P_2)^{\alpha_2}$
$\mathcal{PP} : (P_1, bP_1, (U_{1,j})_{j=0}^m, W_1, g_T)$

All the secret scalars present in the $\mathcal{MSK}$ are known. $\mathscr{B}_1$ can thus create normal keys. However, $\mathscr{B}_1$'s lack of knowledge of the scalar $b$ or its encoding in $\mathbb{G}_2$ does not allow it to create semi-functional keys.

**Key Extraction Phases 1 & 2:** $\mathscr{B}_1$ answers all of $\mathscr{A}$'s queries with normal keys generated by the $\mathcal{IBBE}_1$.$\mathsf{KeyGen}$ algorithm.

**Challenge:** $\mathscr{A}$ sends a challenge set $\widehat{S} = \{\widehat{\mathsf{id}}_1, \ldots, \widehat{\mathsf{id}}_{\widehat{\ell}}\}$. $\mathscr{B}$ sets $(\widehat{\mathsf{Hdr}}, K_0)$ as follows.

For $i = 1, \ldots, \widehat{\ell}$, choose $\widehat{\mathsf{tag}}_i \xleftarrow{\text{U}} \mathbb{Z}_p$,<br>
$K_0 = e(sbP_1, P_2)^{\alpha_2} e((s+\mu)P_1, P_2)^{\alpha_1} = g_T^s e(P_1, P_2)^{\alpha_1\mu}$,<br>
$\widehat{C}_1 = (s+\mu)P_1 = sP_1 + \mu P_1$,<br>
$\widehat{C}_2 = sbP_1$,

For $i = 1, \ldots, \widehat{\ell}$,

$$\begin{aligned}\widehat{C}_{3,i} &= (\textstyle\sum_{j=0}^m \Delta_j (\widehat{\mathsf{id}}_i)^j + \widehat{\mathsf{tag}}_i \cdot \Delta)(sbP_1) + (\textstyle\sum_{j=0}^m e_j (\widehat{\mathsf{id}}_i)^j + \widehat{\mathsf{tag}}_i \cdot c)(s+\mu)P_1 \\ &= (\textstyle\sum_{j=0}^m (\widehat{\mathsf{id}}_i)^j (\Delta_j b + e_j) + \widehat{\mathsf{tag}}_i (\Delta b + c))(sP_1) + (\textstyle\sum_{j=0}^m e_j (\widehat{\mathsf{id}}_i)^j + \widehat{\mathsf{tag}}_i \cdot c)(\mu P_1) \\ &= s(\textstyle\sum_{j=0}^m (\widehat{\mathsf{id}}_i)^j U_{1,j} + \widehat{\mathsf{tag}}_i W_1) + \mu(\textstyle\sum_{j=0}^m e_j (\widehat{\mathsf{id}}_i)^j + \widehat{\mathsf{tag}}_i \cdot c)P_1.\end{aligned}$$

$\mathscr{B}_1$ sets $\widehat{\mathsf{Hdr}} = (\widehat{C}_1, \widehat{C}_2, (\widehat{C}_{3,i}, \widehat{\mathsf{tag}}_i)_{i=1}^{\widehat{\ell}})$. It then samples $K_1 \xleftarrow{\text{U}} \mathbb{G}_T$, $\beta \xleftarrow{\text{U}} \{0,1\}$ and returns the pair $(\widehat{\mathsf{Hdr}}, K_\beta)$ to $\mathscr{A}$. Observe that $(\widehat{\mathsf{Hdr}}, K_0)$ is normal if $\mu = 0$ and semi-functional when $\mu \in_{\text{U}} \mathbb{Z}_p$.

**Guess:** $\mathscr{A}$ outputs its guess $\beta'$ and halts.

$\mathscr{B}$ returns 1 if $\mathscr{A}$'s guess is correct i.e., $\beta = \beta'$; otherwise $\mathscr{B}_1$ returns 0. The advantage of $\mathscr{B}_1$ in solving the DDH1* instance is given by

$$\begin{aligned}\mathsf{Adv}_{\mathcal{G}}^{\text{DDH1}^*}(\mathscr{B}_1) &= |\Pr[\mathscr{B}_1 \text{ returns } 1 | \mu = 0] - \Pr[\mathscr{B}_1 \text{ returns } 1 | \mu \in_{\text{U}} \mathbb{Z}_p]| \\ &= |\Pr[\beta = \beta' | \mu = 0] - \Pr[\beta = \beta' | \mu \in_{\text{U}} \mathbb{Z}_p]| \\ &= |\Pr[\mathscr{A} \text{ wins in } \mathsf{G}_{real}] - \Pr[\mathscr{A} \text{ wins in } \mathsf{G}_0]| \\ &= |\Pr[X_{real}] - \Pr[X_0]|.\end{aligned}$$

Since $\mathsf{Adv}_{\mathcal{G}}^{\text{DDH1}^*}(\mathscr{B}_1) \leq \varepsilon_{\text{DDH1}^*} \leq \varepsilon_{\text{DDH1}}$ (from Propostion 2.1), we have $|\Pr[X_{real}] - \Pr[X_0]| \leq \varepsilon_{\text{DDH1}}$. □

**Lemma 3.2.** $|\Pr[X_{k-1}] - \Pr[X_k]| \leq \varepsilon_{\text{DDH2}}$.

*Proof.* $\mathscr{B}_2$ is given an instance $(\mathcal{G}, rP_2, cP_2, (rc+\gamma)P_2)$ of DDH2 and has to decide whether $\gamma = 0$ or $\gamma \in_{\text{U}} \mathbb{Z}_p$. It simulates the game as described below.

**Setup:** Pick scalars $\alpha_1, \alpha_2', \Delta', (e_{j,1}, e_{j,2}, \Delta_j')_{j=0}^m \xleftarrow{\text{U}} \mathbb{Z}_p$ and $b \xleftarrow{\text{U}} \mathbb{Z}_p^\times$ and (implicitly) set

$$\alpha_2 = \frac{\alpha_2' - \alpha_1}{b}, \quad \Delta = \frac{\Delta' - c}{b},$$

$$e_j = e_{j,1} + ce_{j,2}, \quad \Delta_j = \frac{\Delta_j' - e_j}{b} \quad \text{for } j = 0, \ldots, m.$$

Parameters are generated as follows.

$U_{1,j} = \Delta_j' P_1$ for $j = 0, \ldots, m$, $W_1 = -\Delta' P_1$,<br>
$g_T = e(P_1, P_2)^{\alpha_2'}$<br>
$\mathcal{PP} : (P_1, bP_1, (U_{1,j})_{j=0}^m, W_1, g_T)$

The elements $\Delta, \Delta_j, e_j$ that are part of the $\mathcal{MSK}$ are not available to $\mathscr{B}_2$. Even without these, $\mathscr{B}_2$ can generate keys as explained in the simulation of the key generation phases.

**Key Extraction Phases:** $\mathscr{A}$ queries on identities $\mathsf{id}_1, \mathsf{id}_2, \ldots, \mathsf{id}_q$. $\mathscr{B}$ responds to the $\nu$-th query ($\nu \in [1, q]$) considering three cases.

**Case 1:** $\nu > k$

$\mathscr{B}_2$ returns a normal key, $\mathcal{SK}_{\mathsf{id}_\nu} = (D_1, \ldots, D_5)$. The master secret is not completely available to $\mathscr{B}_2$ and hence the $\mathcal{IBBE}_1$.$\mathsf{KeyGen}$ needs a modification. The components of the key are computed as shown below.

$$r_\nu \xleftarrow{\text{U}} \mathbb{Z}_p,$$

$$D_1 = r_\nu P_2, \;\; D_2 = r_\nu (cP_2),$$

$$\begin{aligned} D_3 &= \left(\alpha_1 + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j e_{j,1}\right)\right) P_2 + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j e_{j,2}\right) (cP_2) \\ &= \left(\alpha_1 + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j e_j\right)\right) P_2, \end{aligned}$$

$$D_4 = b^{-1} r_\nu (\Delta' P_2 - cP_2) = r_\nu \left(\frac{\Delta' - c}{b}\right) P_2 = r_\nu \Delta P_2,$$

$$\begin{aligned} D_5 &= b^{-1} \left(\alpha_2' - \alpha_1 + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j (\Delta_j' - e_{j,1})\right)\right) P_2 - b^{-1} r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j e_{j,2}\right) (cP_2) \\ &= b^{-1} \left(\alpha_2' - \alpha_1 + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j (\Delta_j' - e_{j,1} - ce_{j,2})\right)\right) P_2 \\ &= \left(\frac{\alpha_2' - \alpha_1}{b} + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j \left(\frac{\Delta_j' - e_j}{b}\right)\right)\right) P_2 \\ &= \left(\alpha_2 + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j \Delta_j\right)\right) P_2. \end{aligned}$$

**Case 2:** $\nu < k$

In this case, $\mathscr{B}_2$ first creates a normal key $\mathcal{SK}_{\mathsf{id}_\nu}$ and runs $\mathcal{IBBE}_1$.SFKeyGen on $\mathcal{SK}_{\mathsf{id}_\nu}$. This is possible because the only scalar used in $\mathcal{IBBE}_1$.SFKeyGen is $b$ which is known to $\mathscr{B}_2$.

**Case 3:** $\nu = k$

$\mathscr{B}_2$ embeds the DDH2 instance (consisting of $rP_2, cP_2, (rc+\gamma)P_2$) in the key $\mathcal{SK}_{\mathsf{id}_k} = (D_1, \ldots, D_5)$ for $\mathsf{id}_k$ by generating the components as shown below.

$$D_1 = rP_2, \;\; D_2 = (rc + \gamma)P_2,$$

$$\begin{aligned} D_3 &= \alpha_1 P_2 + \left(\sum_{j=0}^{m} (\mathsf{id}_k)^j e_{j,1}\right) (rP_2) + \left(\sum_{j=0}^{m} (\mathsf{id}_k)^j e_{j,2}\right) (rc + \gamma) P_2 \\ &= \alpha_1 P_2 + r \left(\sum_{j=0}^{m} (\mathsf{id}_k)^j (e_{j,1} + ce_{j,2})\right) P_2 + \gamma \left(\sum_{j=0}^{m} (\mathsf{id}_k)^j e_{j,2}\right) P_2 \\ &= \left(\alpha_1 + r \left(\sum_{j=0}^{m} (\mathsf{id}_k)^j e_j\right)\right) P_2 + \gamma \left(\sum_{j=0}^{m} (\mathsf{id}_k)^j e_{j,2}\right) P_2, \end{aligned}$$

$$D_4 = b^{-1}(\Delta' r P_2 - (rc + \gamma)P_2) = r \left(\frac{\Delta' - c}{b}\right) P_2 - \left(\frac{\gamma}{b}\right) P_2 = r\Delta P_2 - \left(\frac{\gamma}{b}\right) P_2,$$

$$D_5 = b^{-1} \left(\sum_{j=0}^{m} (\mathsf{id}_k)^j (\Delta_j' - e_{j,1})\right) (rP_2) - b^{-1} \left(\sum_{j=0}^{m} (\mathsf{id}_k)^j e_{j,2}\right) (rc + \gamma) P_2$$

$$
\begin{aligned}
&= b^{-1}r\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j(\Delta'_j - e_j)\right)P_2 - b^{-1}\gamma\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j e_{j,2}\right)P_2\\
&= r\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j\left(\frac{\Delta'_j - e_j}{b}\right)\right)P_2 - \left(\frac{\gamma}{b}\right)\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j e_{j,2}\right)P_2\\
&= r\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j\Delta_j\right)P_2 - \left(\frac{\gamma}{b}\right)\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j e_{j,2}\right)P_2,
\end{aligned}
$$

implicitly setting $r_k = r$ and $\gamma_k = \gamma$. When $\gamma = 0$, $\mathcal{SK}_{\mathsf{id}_k}$ is normal; otherwise, it is semi-functional with $\pi_k = \sum_{j=0}^{m}(\mathsf{id}_k)^j e_{j,2}$ set implicitly.

**Challenge:** $\mathscr{B}_2$ obtains the challenge set $\widehat{S} = \{\widehat{\mathsf{id}}_1, \ldots, \widehat{\mathsf{id}}_{\widehat{\ell}}\}$ from $\mathscr{A}$. It then picks $s, \mu \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and generates semi-functional key $K_0$ and header $\widehat{\mathsf{Hdr}} = (\widehat{C}_1, \widehat{C}_2, (\widehat{C}_{3,i}, \widehat{\mathsf{tag}}_i)_{i=1}^{\widehat{\ell}})$ as follows.

$$
\begin{aligned}
&\widehat{\mathsf{tag}}_i = -\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j e_{j,2},\\
&K_0 = g_T^s \cdot e(P_1, P_2)^{\alpha_1\mu},\\
&\widehat{C}_1 = sP_1 + \mu P_1,\\
&\widehat{C}_2 = sbP_1,\\
&\text{For } i = 1, \ldots, \widehat{\ell},\\
&\widehat{C}_{3,i} = s\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j U_{1,j} + \widehat{\mathsf{tag}}_i W_1\right) + \mu\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j e_{j,1}\right)P_1\\
&\quad = s\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j U_{1,j} + \widehat{\mathsf{tag}}_i W_1\right)\\
&\qquad + \mu\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j(e_{j,1} + ce_{j,2}) + \widehat{\mathsf{tag}}_i \cdot c\right)P_1 - \mu\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j ce_{j,2}\right)P_1 - \widehat{\mathsf{tag}}_i \cdot c\mu P_1\\
&\quad = s\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j U_{1,j} + \widehat{\mathsf{tag}}_i W_1\right)\\
&\qquad + \mu\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j e_j + \widehat{\mathsf{tag}}_i \cdot c\right)P_1 - c\mu\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j e_{j,2} + \widehat{\mathsf{tag}}_i\right)P_1\\
&\quad = s\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j U_{1,j} + \widehat{\mathsf{tag}}_i W_1\right) + \mu\left(\textstyle\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j e_j + \widehat{\mathsf{tag}}_i \cdot c\right)P_1.
\end{aligned}
$$

The last step follows due to the fact that $\widehat{\mathsf{tag}} = -\sum_{j=0}^{m}(\widehat{\mathsf{id}}_i)^j e_{j,2}$. $\mathscr{B}_2$ chooses $K_1 \xleftarrow{\mathrm{U}} \mathbb{G}_T$, $\beta \xleftarrow{\mathrm{U}} \{0, 1\}$ and returns $(\widehat{\mathsf{Hdr}}, K_\beta)$ to $\mathscr{A}$. Note that $\widehat{\mathsf{Hdr}}$ and $K_0$ are properly formed. Also, this is the only way $\mathscr{B}_2$ can generate a semi-functional header-key pair since no encoding of $c$ is available in the group $\mathbb{G}_1$. An implication is that $\mathscr{B}_2$ can only create a nominally semi-functional header component with index $i$ for a set of intended recipients containing $\mathsf{id}_k$ as the $i$-th identity. This is because the relation $\mathsf{tag}_i = -\pi_k$ will hold. This provides no information to $\mathscr{B}_2$ about the semi-functionality of $\mathcal{SK}_{\mathsf{id}_k}$.

**Guess:** $\mathscr{A}$ returns its guess $\beta'$ of $\beta$.

$\mathscr{B}_2$ outputs 1 if $\mathscr{A}$ wins and 0 otherwise. Also, $\mathscr{B}_2$ simulates $\mathsf{G}_{k-1}$ if $\gamma = 0$ and $\mathsf{G}_k$ if $\gamma \in_{\mathrm{U}} \mathbb{Z}_p$. Therefore, the advantage of $\mathscr{B}_2$ in solving the DDH2 instance is given by

$$
\begin{aligned}
\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH2}}(\mathscr{B}_2) &= |\Pr[\mathscr{B}_2 \text{ returns } 1 | \gamma = 0] - \Pr[\mathscr{B}_2 \text{ returns } 1 | \gamma \in_{\mathrm{U}} \mathbb{Z}_p]|\\
&= |\Pr[\beta = \beta' | \mu = 0] - \Pr[\beta = \beta' | \mu \in_{\mathrm{U}} \mathbb{Z}_p]|\\
&= |\Pr[\mathscr{A} \text{ wins in } \mathsf{G}_{k-1}] - \Pr[\mathscr{A} \text{ wins in } \mathsf{G}_k]|\\
&= |\Pr[X_{k-1}] - \Pr[X_k]|.
\end{aligned}
$$

It now follows that $|\Pr[X_{k-1}] - \Pr[X_k]| \leq \varepsilon_{\mathrm{DDH2}}$ from the fact that $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH2}}(\mathscr{B}) \leq \varepsilon_{\mathrm{DDH2}}$ for all $t$-time adversaries $\mathscr{B}$. What remains is to show that all the information provided to the adversary have the correct distribution. The scalars $b, \alpha_1, \alpha'_2, \Delta', (e_{j,1}, e_{j,2}, \Delta'_j)_{j=0}^{m}$ chosen by $\mathscr{B}_2$ and $r, c, \gamma$ from the instance are uniformly and independently distributed in their respective domains. These scalars determine the distribution of the following quantities.

- $\alpha_2, \Delta$
- $(e_j)_{j=0}^m$ and hence $(\Delta_j)_{j=0}^m$
- $r_k, \gamma_k$
- $\pi_k$
- $\widehat{\mathsf{tag}}_1, \dots, \widehat{\mathsf{tag}}_{\widehat{\ell}}$

$(\alpha_2, \Delta)$ are uniquely determined by $(\alpha_2', \Delta')$. Scalars $r_k$, $\gamma_k$ have the correct distribution since they are set to $r$, $\gamma$ respectively. Also, all other information is independent of $r, \gamma$. We will now argue that $\pi_k$ and $\widehat{\mathsf{tag}}_1, \dots, \widehat{\mathsf{tag}}_{\widehat{\ell}}$ are properly distributed. They are given by the following equation.

$$\begin{pmatrix} \pi_k \\ \widehat{\mathsf{tag}}_1 \\ \widehat{\mathsf{tag}}_2 \\ \vdots \\ \widehat{\mathsf{tag}}_{\widehat{\ell}} \end{pmatrix} = \begin{pmatrix} 1 & \mathsf{id}_k & (\mathsf{id}_k)^2 & \cdots & (\mathsf{id}_k)^m \\ 1 & \widehat{\mathsf{id}}_1 & (\widehat{\mathsf{id}}_1)^2 & \cdots & (\widehat{\mathsf{id}}_1)^m \\ 1 & \widehat{\mathsf{id}}_2 & (\widehat{\mathsf{id}}_2)^2 & \cdots & (\widehat{\mathsf{id}}_2)^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \widehat{\mathsf{id}}_{\widehat{\ell}} & (\widehat{\mathsf{id}}_{\widehat{\ell}})^2 & \cdots & (\widehat{\mathsf{id}}_{\widehat{\ell}})^m \end{pmatrix} \begin{pmatrix} e_{0,2} \\ e_{1,2} \\ \vdots \\ e_{m,2} \end{pmatrix} \tag{1}$$

One can make the following observations.

- $\mathsf{id}_k, \widehat{\mathsf{id}}_1, \dots, \widehat{\mathsf{id}}_{\widehat{\ell}}$ are all distinct since $\mathsf{id}_k \notin \widehat{S}$. Also $\widehat{\ell} \leq m$. Hence the above matrix of order $(\widehat{\ell}+1) \times (m+1)$ over $\mathbb{Z}_p$ is a Vandermonde matrix and has rank $\widehat{\ell}+1$.
- $e_{0,2}, e_{1,2}, \dots, e_{m,2}$ are information theoretically hidden from $\mathscr{A}$ and also chosen uniformly and independently over $\mathbb{Z}_p$.

From these observations, it follows that $\pi_k, \widehat{\mathsf{tag}}_1, \dots, \widehat{\mathsf{tag}}_{\widehat{\ell}}$ are uniformly and independently distributed in $\mathscr{A}$'s view.

The scalars $(\Delta_j)_{j=0}^m$ are uniquely determined by $(\Delta_j')_{j=0}^m$ and $(e_j)_{j=0}^m$. So all that we need to show is that the quantities $e_j = e_{j,1} + ce_{j,2}$ for $j \in [0, m]$ have the right distribution conditioned on $\pi_k$ and tags being determined by $(e_{j,2})_{j=0}^m$. This follows from the fact that $e_{j,1}$'s are uniformly and independently distributed in $\mathbb{Z}_p$ thus making the $e_j$'s uniform random quantities in $\mathbb{Z}_p$. □

**Lemma 3.3.** $\Pr[X_q] = \Pr[X_{final}]$ *and* $|\Pr[X_{final}] - 1/2| \leq 1/p$.

*Proof.* In $\mathsf{G}_q$, all the user keys returned to $\mathscr{A}$ are semi-functional and so is the challenge header and key. We now modify the setup and key extraction phases so that the modification results in $\mathsf{G}_{final}$ and then argue that the resulting game is indistinguishable from $\mathsf{G}_q$ except for probability $q/p$.

**Setup:** Pick scalars $\alpha_1, \alpha_2', \Delta', c, (\Delta_j', e_j)_{j=0}^m \xleftarrow{\text{U}} \mathbb{Z}_p$ and $b \xleftarrow{\text{U}} \mathbb{Z}_p^\times$ and compute parameters as:

$U_{1,j} = \Delta_j' P_1$ for $j = 0, \dots, m$, $W_1 = \Delta' P_1$,

$g_T = e(P_1, P_2)^{\alpha_2'}$

$\mathcal{PP} : (P_1, bP_1, (U_{1,j})_{j=0}^m, W_1, g_T)$

setting

$$\alpha_2 = \frac{\alpha_2' - \alpha_1}{b}, \quad \Delta = \frac{\Delta' - c}{b},$$

$$\Delta_j = \frac{\Delta_j' - e_j}{b} \quad \text{for } j = 0, \dots, m.$$

Although $\alpha_1$ is sampled during setup, it has no effect on the distribution $g_T$ and hence that of $\mathcal{PP}$. This is because $g_T$ is created using $\alpha_2'$ which is chosen independent of $\alpha_1$.

**Key Extraction:** On a key extract query for $\mathsf{id}$, choose $r, \pi', \gamma \xleftarrow{\text{U}} \mathbb{Z}_p$, and compute the individual components as follows.

$$D_1 = rP_2,\ \ D_2 = rcP_2 + \gamma P_2,\ \ D_3 = \pi' P_2 + r\left(\sum_{j=0}^{m} (\mathsf{id})^j e_j\right) P_2,$$

$$D_4 = r\left(\frac{\Delta' - c}{b}\right) P_2 - \left(\frac{\gamma}{b}\right) P_2,$$

$$D_5 = \left(\frac{\alpha_2' - \pi'}{b}\right) + r\left(\sum_{j=0}^{m} (\mathsf{id})^j \Delta_j\right) P_2.$$

We define $\pi$ in the following manner: if $\gamma = 0$, choose $\pi$ independently and uniformly at random from $\mathbb{Z}_p$; and if $\gamma \neq 0$, then set $\pi = (\pi' - \alpha_1)/\gamma$. In both cases, $\pi$ is uniformly distributed over $\mathbb{Z}_p$ and can be shown to be independent of $\alpha_1$ and all other scalars. Note that this manner of defining $\pi$ ensures that $D_3$ and $D_5$ have the proper semi-functional forms. We show below that $D_5$ is indeed well-formed in this sense.

$$\begin{aligned} D_5 &= \left(\frac{\alpha_2' - \pi'}{b}\right) P_2 + r\left(\sum_{j=0}^{m} (\mathsf{id})^j \Delta_j\right) P_2 \\ &= \left(\frac{\alpha_2' - \alpha_1 - \gamma\pi}{b}\right) P_2 + r\left(\sum_{j=0}^{m} (\mathsf{id})^j \Delta_j\right) P_2 \\ &= \left(\frac{\alpha_2' - \alpha_1}{b}\right) P_2 + r\left(\sum_{j=0}^{m} (\mathsf{id})^j \Delta_j\right) P_2 - \left(\frac{\gamma\pi}{b}\right) P_2 \\ &= \left(\alpha_2 + r\left(\sum_{j=0}^{m} (\mathsf{id})^j \Delta_j\right)\right) P_2 - \left(\frac{\gamma\pi}{b}\right) P_2. \end{aligned}$$

Furthermore, $D_3$ and $D_5$ are generated using $\pi'$ which is chosen independent of $\alpha_1$, thus making the key independent of $\alpha_1$.

**Challenge:** The challenge header and $K_0$ for the challenge privileged users' set $\widehat{S} = \{\mathsf{id}_1, \ldots, \mathsf{id}_{\widehat{\ell}}\}$ are computed as:

$s, \mu \xleftarrow{\text{U}} \mathbb{Z}_p$, $(\mathsf{tag}_i)_{i=1}^{\widehat{\ell}} \xleftarrow{\text{U}} \mathbb{Z}_p$,
$K_0 = g_T^s \cdot e(P_1, P_2)^{\alpha_1 \mu}$,
$\widehat{C}_1 = sP_1 + \mu P_1$,
$\widehat{C}_2 = sbP_1$,
For $i = 1, \ldots, \widehat{\ell}$,
$\widehat{C}_{3,i} = s\left(\sum_{j=0}^{m} (\widehat{\mathsf{id}}_i)^j \Delta_j' + \widehat{\mathsf{tag}}_i \Delta'\right) P_1 + \mu\left(d + \sum_{j=0}^{m} (\widehat{\mathsf{id}}_i)^j e_j + \widehat{\mathsf{tag}}_i \cdot c\right) P_1$.

From this it follows that the adversary's view in Game $\mathsf{G}_q$ is identical to the view in Game $\mathsf{G}_{final}$ and so $\Pr[X_q] = \Pr[X_{final}]$ which proves the first part of the lemma.

Recall that $X_{final}$ denotes the event that the adversary wins in Game $\mathsf{G}_{final}$ which is the event that the adversary's guess $\beta'$ equals $\beta$.

$K_0$ is computed as $g_T^s \cdot e(P_1, P_2)^{\alpha_1 \mu}$ where $\alpha_1$ is independent of all other scalars and has not been used to compute any other component either of the public parameters, or the decryption keys of the challenge ciphertext. So, if $\mu \neq 0$, then $K_0$ is uniformly distributed and is independent of all other components. Note that $K_1$ is uniformly distributed and is independent of all other components. So, if $\mu \neq 0$, then adversary's view for $\beta = 0$ is the same as that for $\beta = 1$. This in particular means that $\Pr[\beta' = 0 | \beta = 0, \mu \neq 0] =$

$\Pr[\beta' = 0|\beta = 1, \mu \neq 0]$. We now compute as follows.

$$\begin{aligned} \Pr[X_{final}] &= \Pr[\beta' = \beta] \\ &\leq 1/p + \Pr[\beta' = \beta|\mu \neq 0] \\ &\leq 1/p + \Pr[\beta' = 0, \beta = 0|\mu \neq 0] + \Pr[\beta' = 1, \beta = 1|\mu \neq 0] \\ &\leq 1/p + \frac{1}{2}\left(\Pr[\beta' = 0|\beta = 0, \mu \neq 0] + \Pr[\beta' = 1|\beta = 1, \mu \neq 0]\right) \\ &\leq 1/p + \frac{1}{2}\left(\Pr[\beta' = 0|\beta = 1, \mu \neq 0] + \Pr[\beta' = 1|\beta = 1, \mu \neq 0]\right) \\ &= 1/p + 1/2. \end{aligned}$$

This proves the second part of the lemma.

□

# 4 Towards Shorter Headers Without Random Oracles

The header size in $\mathcal{IBBE}_1$ is $(\ell + 2)N_1 + \ell N_p$ for a recipient set of size $\ell$ ($\leq m$). As discussed earlier, we cannot do much with the identity hashes and neither can the tags be completely eliminated. One way of tackling the tags is to use a random oracle as has also been mentioned earlier. The question that we address here is whether the issue of increase in the ciphertext size due to the use of tags can be alleviated without resorting to random oracles.

In this section, we provide an answer to this question which results in a trade-off between the number of tags and the number of session key encapsulations. The resulting scheme, which we call $\mathcal{IBBE}_2$, operates as follows. Partition the privileged users' set and encapsulate the session key separately to each subset in the partition by applying the encapsulation algorithm of $\mathcal{IBBE}_1$. These separate encapsulations are not completely independent. The tags are reused across encapsulations. Below, we provide an overview of the scheme followed by the formal details.

Let the maximum size of the privileged users' set be $m = m_1 m_2$. Initialise an $\mathcal{IBBE}_1$ system with $m_2$ as the input to the $\mathsf{Setup}$ algorithm. Suppose we want to encrypt to a set $S$ of size $\ell \leq m$.

1. Express $\ell$ as $\ell = (\ell_1 - 1)m_2 + \ell_2$ where $1 \leq \ell_1 \leq m_1$ and $1 \leq \ell_2 \leq m_2$.
2. Partition $S$ into $\ell_1$ disjoint subsets $S_1, \ldots, S_{\ell_1}$ so that $|S_j| = m_2$ for $j = 1, \ldots \ell_1 - 1$ and $|S_{\ell_1}| = \ell_2$.
3. Choose random tags $\mathsf{tag}_1, \ldots, \mathsf{tag}_{m_2}$ from $\mathbb{Z}_p$. (We need $m_2$ tags since each subset $S_j$ is of size at most $m_2$.)
4. Run $\mathcal{IBBE}_1.\mathsf{Encap}$ on each $S_j$ (for $j \in [1, \ell_1]$) separately with the tags set to $\mathsf{tag}_1, \ldots, \mathsf{tag}_{m_2}$.

This results in $\ell_1$ $\mathcal{IBBE}_1$ headers (referred to as *sub-headers*) with each sub-header consisting of at most $m_2$ elements of $\mathbb{G}_1$. The $\mathcal{IBBE}_2$ header consists of these sub-headers and the $m_2$ tags used to construct all the $\ell_1$ sub-headers in addition to $\ell_1$ elements of $\mathbb{G}_T$ each masking the session key.

The above idea is made concrete as the scheme

$$\mathcal{IBBE}_2 = (\mathcal{IBBE}_2.\mathsf{Setup}, \mathcal{IBBE}_2.\mathsf{Encrypt}, \mathcal{IBBE}_2.\mathsf{KeyGen}, \mathcal{IBBE}_2.\mathsf{Decrypt})$$

which is defined in Figure 2. The encapsulation algorithm makes the call $\mathcal{IBBE}_1.\mathsf{Encap}(\mathcal{PP}', S_j; s_j, \mathsf{tag}_1, \ldots, \mathsf{tag}_{m_2})$. Recall from Section 2.1 that the notation $\mathcal{A}(\cdot; R)$ denotes running the probabilistic algorithm $\mathcal{A}(\cdot)$ with its random bits set to $R$. Also, recall that in the description of $\mathcal{IBBE}_1.\mathsf{Encap}()$, the randomiser $s$ and the tags are chosen independently and uniformly at random and these constitute the entire random choices of the algorithm. So, the encapsulation algorithm of $\mathcal{IBBE}_2$ runs the encapsulation algorithm of $\mathcal{IBBE}_1$ for particular choices of its internal randomness.

**Correctness.** It is straightforward to verify that the correctness of decapsulation follows from that of $\mathcal{IBBE}_1$.

Figure 2: Construction of $\mathcal{IBBE}_2$.

```
$\mathcal{IBBE}_2.\mathsf{Setup}(\kappa, m)$

1. Let $(\mathcal{PP}', \mathcal{MSK}') \xleftarrow{\mathrm{R}} \mathcal{IBBE}_1.\mathsf{Setup}(\kappa, m_2)$.
Define $\mathcal{PP} = (\mathcal{PP}', m_2)$.
Define $\mathcal{MSK} = \mathcal{MSK}'$.
```

```
$\mathcal{IBBE}_2.\mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id})$

1. $\mathcal{SK}_{\mathsf{id}} \xleftarrow{\mathrm{R}} \mathcal{IBBE}_1.\mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id})$.
Return $\mathcal{SK}_{\mathsf{id}}$.
```

```
$\mathcal{IBBE}_2.\mathsf{Encap}(\mathcal{PP}, S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\})$.

1. Write $\ell = (\ell_1 - 1)m_2 + \ell_2$
     with $1 \leq \ell_1 \leq m_1$ and $1 \leq \ell_2 \leq m_2$.
2. Partition $S$ into $\ell_1$ disjoint subsets $S_1, S_2, \ldots, S_{\ell_1}$
     where $|S_j| = m_2$ for $j \in [1, \ell_1 - 1]$ and $|S_{\ell_1}| = \ell_2$.
3. Choose $(s_j)_{j=1}^{\ell_1}, (\mathsf{tag}_i)_{i=1}^{m_2} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.
4. For $j = 1, \ldots, \ell_1$
     $(\mathsf{Hdr}_j, K_j) \longleftarrow \mathcal{IBBE}_1.\mathsf{Encap}(\mathcal{PP}', S_j; s_j, (\mathsf{tag}_i)_{i=1}^{m_2})$.
5. Choose $K' \xleftarrow{\mathrm{U}} \mathbb{G}_T$.
6. For $j = 1$ to $\ell_1$
     Compute $C_{0,j} = K' \cdot K_j$.
7. Set $\vec{\mathsf{Hdr}} = ((\mathsf{Hdr}_j, C_{0,j})_{j=1}^{\ell_1}, (\mathsf{tag}_i)_{i=1}^{m_2})$.
Return $(\vec{\mathsf{Hdr}}, K')$.
```

```
$\mathcal{IBBE}_2.\mathsf{Decap}(\mathcal{PP}, S, \mathsf{id}, \mathcal{SK}_{\mathsf{id}}, \mathsf{Hdr})$

1. Parse $S$ as $(S_1, \ldots, S_{\ell_1})$.
2. If $\mathsf{id} \in S_j$ for some $j \in [1, \ell_1]$
3.    Let $\mathcal{P} = (\mathcal{PP}', S_j, \mathsf{id}, \mathcal{SK}_{\mathsf{id}}, \mathsf{Hdr}_j, (\mathsf{tag}_i)_{i=1}^{m_2})$.
4.    $K_j = \mathcal{IBBE}_1.\mathsf{Decap}(\mathcal{P})$.
5.    Compute $K' = C_{0,j} \cdot K_j^{-1}$.
6. Else $K' = \perp$.
Return $K'$.
```

**Masked copies of the session key:** The message is encrypted using the session key $K'$ and $C_{0,j}$, $1 \leq j \leq \ell_1$, are the masked copies of $K'$. In the above description, $K'$ is from $\mathbb{G}_T$ since this is convenient for the security analysis. In practice, however, $K'$ will be the key for a DEM and hence will be a $\kappa$-bit string, where $\kappa$ is the security parameter. In this case, the quantities $C_{0,j}$ will be generated as $\mathsf{KDF}(K_j) \oplus K'$, where $\mathsf{KDF}$ is a key derivation function which maps an element of $\mathbb{G}_T$ to a $\kappa$-bit string. As a result, $C_{0,1}, \ldots, C_{0,\ell_1}$ consists of $\ell_1$ $\kappa$-bit strings. While considering the efficiency of $\mathcal{IBBE}_2$, we will consider the $C_{0,j}$'s to be $\kappa$-bit strings. For the security analysis, on the other hand, we will proceed with considering the $C_{0,j}$'s to be elements of $\mathbb{G}_T$. Modifying this security analysis to consider $C_{0,j}$'s to be $\kappa$-bit strings will require considering the security of $\mathsf{KDF}$. This is quite routine and hence we skip it.

**Header size for $\mathcal{IBBE}_2$:** The total size of the $\mathcal{IBBE}_2$ header is $(\ell + 2\ell_1)N_1 + m_2 N_p + \ell_1 \kappa$ (assuming $C_{0,j}$'s to be $\kappa$-bit strings). In comparison, the header size for $\mathcal{IBBE}_1$ is $(\ell + 2)N_1 + \ell N_p$. A reasonable estimate of the group sizes is $N_1 = 2N_p$ and $N_p = 2\kappa$. Also, assume that $m_1$ and $m_2$ are around $\sqrt{m}$. For small $\ell$, the header sizes of the two IBBE schemes are comparable. For $\ell$ around $m$, the header size of $\mathcal{IBBE}_2$ is smaller for $m \geq 25$.

**Generating tags using a random oracle.** As in the case of $\mathcal{IBBE}_1$, it is possible to construct a variant $\mathcal{IBBE}_2^{\mathrm{RO}}$ of $\mathcal{IBBE}_2$ that is adaptively secure with random oracles. The tags used in encryption are generated using a random oracle as in $\mathcal{IBBE}_1^{\mathrm{RO}}$. The construction $\mathcal{IBBE}_2^{\mathrm{RO}}$ can be obtained by just replacing $\mathcal{IBBE}_1$ by $\mathcal{IBBE}_1^{\mathrm{RO}}$ in the description of $\mathcal{IBBE}_2$ above. Moreover, $\mathcal{IBBE}_2^{\mathrm{RO}}$ can be shown to be secure based on the assumption that $\mathcal{IBBE}_1^{\mathrm{RO}}$ is secure. The header for $\mathcal{IBBE}_2^{\mathrm{RO}}$ consists of $(\ell + 2\ell_1)$ elements of $\mathbb{G}_1$ and $\ell_1$ $\kappa$-bit masked versions of the session key and a single $\kappa$-bit quantity from which the $m_2$ tags are generated using the random oracle. In contrast, the header for $\mathcal{IBBE}_1^{\mathrm{RO}}$ consists of $(\ell + 2)$ elements of $\mathbb{G}_1$ and a single $\kappa$-bit quantity from which the $m_2$ tags are generated. As a result, the header size for $\mathcal{IBBE}_2^{\mathrm{RO}}$ is greater than the header size for $\mathcal{IBBE}_1^{\mathrm{RO}}$. So, if the tags are to be generated using a hash function, which is modelled as a random oracle, then it is more advantageous to use $\mathcal{IBBE}_1^{\mathrm{RO}}$ than $\mathcal{IBBE}_2^{\mathrm{RO}}$. We note that the PP size of $\mathcal{IBBE}_2^{\mathrm{RO}}$ is lower than that of $\mathcal{IBBE}_1^{\mathrm{RO}}$, but, this is of lesser significance.

**Restriction on the size of the identity set:** As in the case of $\mathcal{IBBE}_1$, in the encapsulation algorithm we have assumed that the number of identities $\ell$ to which the message is to be encrypted is at most $m$. In case $\ell > m$, then the comment made in the context of $\mathcal{IBBE}_1$ also applies for $\mathcal{IBBE}_2$.

## 4.1 Security of $\mathcal{IBBE}_2$

The security of $\mathcal{IBBE}_2$ follows from that of $\mathcal{IBBE}_1$. More precisely, we show that $\mathcal{IBBE}_2$ is secure if $\mathcal{IBBE}_1$ is secure as formalised in the theorem below.

**Theorem 4.1.** *If $\mathcal{IBBE}_1$ is $(\varepsilon, t, q)$-*IND-ID-CPA*-secure then $\mathcal{IBBE}_2$ is $(\varepsilon', t', q)$-*IND-ID-CPA*-secure where $\varepsilon' \le 2m_1\varepsilon$ and $t' = O(m_1 t)$.*

*Proof.* The proof is via a simple hybrid argument over the session key encryptions. Let $\mathscr{A}$ be a $t$-time IND-ID-CPA adversary against $\mathcal{IBBE}_2$. We show how to build IND-ID-CPA adversaries $\mathscr{B}_1, \ldots, \mathscr{B}_{\widehat{\ell_1}}$ (where $\widehat{\ell_1} \le m_1$ is the size of the partition of the challenge set) all running in time $t$ against $\mathcal{IBBE}_1$ such that $\mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{IBBE}_2}(\mathscr{A}) \le \sum_{\nu=1}^{\widehat{\ell_1}} \mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{IBBE}_1}(\mathscr{B}_t)$. Since $\widehat{\ell_1} \le m_1$, the statement of the theorem follows.

Define the following game sequence: $\mathsf{G}_0, \mathsf{G}_1, \ldots, \mathsf{G}_{\widehat{\ell_1}}$ where $\mathsf{G}_0$ is the real ind-cpa game; in $\mathsf{G}_\nu$ ($\nu \in [1, \widehat{\ell_1}]$), the first $\nu$ encryptions of the session key are random and the rest are normally formed. Let $Y_\square$ denote the probability that $\mathscr{A}$ wins in $\mathsf{G}_\square$.
**Transition from $\mathsf{G}_{\nu-1}$ to $\mathsf{G}_\nu$ for $\nu \in [1, \widehat{\ell_1}]$:** $\mathscr{B}_\nu$ receives the public parameters $\mathcal{PP}'$ of $\mathcal{IBBE}_1$ from its challenger and returns $\mathcal{PP} = (\mathcal{PP}, m_2)$ to $\mathscr{A}$. A key extraction query on an identity $\mathsf{id}$ that $\mathscr{A}$ makes is answered with the secret key that $\mathscr{B}_\nu$ receives from its challenger on the same identity. In the challenge phase, $\mathscr{B}_\nu$ receives a set $\widehat{S}$ from $\mathscr{A}$ and paritions it as $(\widehat{S}_1, \ldots, \widehat{S}_{\widehat{\ell_1}})$ with each $|\widehat{S}_j| = m_2$ for $j \in [1, \widehat{\ell}_1 - 1]$ and $|S_{\widehat{\ell_1}}| = \widehat{\ell_2}$. $\mathscr{B}_\nu$ provides $\widehat{S}_\nu$ to its challenger and obtains a pair $(\widehat{\mathsf{Hdr}}, K_\beta)$. It then extracts the tags in $\widehat{\mathsf{Hdr}}$, denoted $(\widehat{\mathsf{tag}}_i)_{i=1}^{\widehat{m_2}}$, picks a random bit $\delta \xleftarrow{\mathrm{U}} \{0,1\}$ and sets

$$(\mathsf{Hdr}_j, K_j) \xleftarrow{\mathrm{R}} \mathcal{IBBE}_1.\mathsf{Encap}(\mathcal{PP}', S_j; (\mathsf{tag}_i)_{i=1}^{\widehat{m_2}}), \text{ for } j \in [1, \widehat{\ell}_1] \setminus \{\nu\},$$

$$K_0', K_1' \xleftarrow{\mathrm{U}} \mathbb{G}_T,$$

$$C_{0,j} \xleftarrow{\mathrm{U}} \mathbb{G}_T \text{ for } j \in [1, \nu - 1], \quad C_{0,j} \leftarrow K_\delta' \cdot K_j \text{ for } j = [\nu + 1, \widehat{\ell}_1],$$

$$\mathsf{Hdr}_\nu = \widehat{\mathsf{Hdr}}, \quad C_{0,\nu} \leftarrow K_\delta' \cdot K_\beta,$$

$$\widehat{\overrightarrow{\mathsf{Hdr}}} = \left( (\mathsf{Hdr}_j, C_{0,j})_{j=1}^{\widehat{\ell}_1}, (\mathsf{tag}_i)_{i=1}^{\widehat{m_2}} \right).$$

$\mathscr{B}_\nu$ returns $\widehat{\overrightarrow{\mathsf{Hdr}}}, K_\delta'$ to $\mathscr{A}$. The adversary $\mathscr{A}$ returns its guess $\delta'$ of $\delta$. $\mathscr{B}_\nu$ sets $\beta' = 1$ if $\delta = \delta'$; else it sets $\beta' = 0$ and returns $\beta'$ to its challenger.

We have

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{IBBE}_1}(\mathscr{B}_\nu) &= \left|\Pr[\beta = \beta'] - \frac{1}{2}\right| \\
&= \left|\Pr[\beta' = 1|\beta = 1]\Pr[\beta = 1] + \Pr[\beta' = 0|\beta = 0]\Pr[\beta = 0] - \frac{1}{2}\right| \\
&= \frac{1}{2}\left|\Pr[\beta' = 1|\beta = 1] - \Pr[\beta' = 1|\beta = 0]\right| \\
&= \frac{1}{2}\left|\Pr[\delta = \delta'|\beta = 1] - \Pr[\delta = \delta'|\beta = 0]\right| \\
&= \frac{1}{2}\left|\Pr[\delta = \delta' \text{ in } \mathsf{G}_\nu] - \Pr[\delta = \delta' \text{ in } \mathsf{G}_{\nu-1}]\right| \\
&= \frac{1}{2}\left|\Pr[Y_\nu] - \Pr[Y_{\nu-1}]\right|.
\end{aligned}$$

Since $\Pr[Y_{\widehat{\ell}_1}] = 1/2$, we have $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{IBBE}_2}(\mathscr{A}) = |\Pr[Y_0] - \Pr[Y_{\widehat{\ell}_1}]| \leq \sum_{\nu=1}^{\widehat{\ell}_1} |\Pr[Y_{\nu-1}] - \Pr[Y_\nu]| = 2\sum_{\nu=1}^{\widehat{\ell}_1} \mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{IBBE}_1}(\mathscr{B}_\nu)$, as required. □ □

# 5 From IB(B)E to PKBE: Dodis-Fazio Revisited

Dodis and Fazio [DF02] described a method to build a public-key broadcast encryption scheme from an identity-based encryption scheme. The core idea behind this conversion is a combinatorial structure called complete subtree (CS) symmetric key revocation scheme introduced by Naor, Naor and Lotspeich [NNL01].

In the CS scheme, the number of users $n$ is assumed to be a power of 2 and the users are organized as the leaves of a complete binary tree $\mathcal{T}$ of height $\log n$. If $v$ is a node of $\mathcal{T}$, define $\mathcal{S}_v$ to be the set of all leaf nodes in the subtree rooted at $v$. Further, let $\mathscr{C}$ be the collection of $\mathcal{S}_v$ for all $v$ in $\mathcal{T}$. A centre assigns keys to subsets in $\mathscr{C}$. During a pre-distribution phase, a user corresponding to a leaf node $u$ receives keys for all subsets in $\mathscr{C}$ which contains $u$. During an actual broadcast, the centre identifies a set of $r$ revoked users. A partition of the other $n-r$ users is created using subsets from $\mathscr{C}$. Suppose the partition consists of $h$ subsets $\mathcal{S}_1, \ldots, \mathcal{S}_h$. The actual message is encrypted using a session key and the session key is then encrypted using the keys corresponding to the $h$ subsets $\mathcal{S}_1, \ldots, \mathcal{S}_h$. The encryptions of the session key constitute the header. It has been shown in [NNL01] that each user has to store $\log n$ keys and the size of the header is at most $r\log(n/r)$.

Dodis and Fazio [DF02] presented a method to combine the CS scheme with an IBE scheme to obtain a PKBE scheme. The idea is as follows. The role of the centre in the CS scheme is played by the PKG of the IBE scheme. Set-up of the PKBE scheme consists of the following steps:

- the PKG runs the $\mathsf{Setup}$ algorithm of an IBE scheme;
- assigns an identity $\mathsf{id}_{\mathcal{S}}$ to each subset $\mathcal{S}$ in the collection $\mathscr{C}$;
- generates corresponding keys $\mathcal{SK}_{\mathsf{id}_{\mathcal{S}}}$ using the $\mathsf{KeyGen}$ algorithm of the IBE scheme;
- provides each user $u$ with $\mathcal{SK}_{\mathsf{id}_{\mathcal{S}}}$ for each $\mathcal{S}$ to which it belongs;
- publishes $\mathcal{PP}$ and the structure $\mathcal{T}$ as the public key of the PKBE scheme.

Here $\mathcal{PP}$ consists of the public parameters of the IBE scheme.

For an actual broadcast, an entity forms a partition of the set of privileged users as in the CS scheme. As before, suppose that the partition consists of $h$ sets $\mathcal{S}_1, \ldots, \mathcal{S}_h$ from $\mathscr{C}$. Let the corresponding identities be $\mathsf{id}_{\mathcal{S}_1}, \ldots, \mathsf{id}_{\mathcal{S}_h}$. As in the CS scheme, the actual message is encrypted using a session key. Using $\mathcal{PP}$, the session key is encrypted $h$ times to the identities $\mathsf{id}_{\mathcal{S}_1}, \ldots, \mathsf{id}_{\mathcal{S}_h}$. These encryptions of the session key form the header. A user in any of the $\mathcal{S}$'s has a secret key $\mathcal{SK}_{\mathsf{id}_{\mathcal{S}}}$ corresponding to $\mathsf{id}_i$. This allows the user to decrypt the corresponding encryption of the session key. The security of the scheme follows from the security of the IBE scheme. A user needs to store $\log n$ IBE keys and a header consists of at most $r\log(n/r)$ IBE encryptions of the session key.

Developing upon the Dodis-Fazio agenda described above, we suggest that the CS scheme be combined with an identity-based *broadcast* encryption scheme to obtain a PKBE scheme. Most of the details will remain unchanged. The only difference will be in the encryption. Suppose as above that $\mathcal{S}_1, \ldots, \mathcal{S}_h$ is the partition of the set of all privileged users and let $\{\mathsf{id}_{\mathcal{S}_1}, \ldots, \mathsf{id}_{\mathcal{S}_h}\}$ be the set of identities corresponding these sets. The Dodis-Fazio transformation mentions that encryptions are to be made individually to these identities. Using an IBBE scheme, on the other hand, one can make a single encryption to the set of identities $\{\mathsf{id}_{\mathcal{S}_1}, \ldots, \mathsf{id}_{\mathcal{S}_h}\}$. Decryption will be as before. The advantage is that the header size will go down. It is routine to argue that the security of the scheme will follow from the security of the IBBE scheme.

To illustrate the trade-offs, suppose that the Dodis-Fazio transformation is instantiated with the JR-IBE-D. The resulting PKBE will have headers consisting of at most $3r\log(n/r), r\log(n/r), r\log(n/r)$ elements from $\mathbb{G}_1, \mathbb{G}_T, \mathbb{Z}_p$ respectively. If on the other hand, we use $\mathcal{IBBE}_1$ as the IBBE scheme to obtain a PKBE scheme from the CS scheme, the maximum header size will be $2 + r\log(n/r), 1, r\log(n/r)$ elements from $\mathbb{G}_1, \mathbb{G}_T, \mathbb{Z}_p$ respectively. The trade-off is that the size of the public parameters will go up. Since public parameters is a static quantity and needs to be downloaded once, the savings in the size of the ciphertext will far outweigh the increase in the size of the public parameters. In arriving at the figures $2 + r\log(n/r), 1, r\log(n/r)$, we have assumed that the number of elements $h$ in the header is at most $m$, the parameter in the $\mathcal{IBBE}_1$ scheme. If, on the other hand, $h$ is more than $m$, then this would lead to a header consisting of encryptions to $\lceil h/m \rceil$ sets of identities as mentioned earlier.

Naor, Naor and Lotspeich [NNL01] described another symmetric key BE scheme called the subset difference (SD) scheme. Dodis and Fazio [DF02] showed how to use a HIBE to convert the SD scheme to a PKBE scheme. This is not relevant in the current context and hence, we do not discuss this any further.

## 6 Conclusion

In this paper, we have presented new IBBE schemes which achieve both theoretically satisfying security (i.e, security against adaptive-identity attacks based on simple assumptions) and practical efficiency at the same time. The new schemes are obtained by developing on the currently known most efficient IBE scheme due to Jutla and Roy [JR13]. As with most prior work, the new schemes are proved secure against chosen-plaintext attacks. It is of interest to obtain efficient variants of these schemes which are secure against chosen ciphertext attacks. Also, actual implementation studies will take the works further along the path of actual deployment.

## References

[AKN07] Michel Abdalla, Eike Kiltz, and Gregory Neven. Generalized key delegation for hierarchical identity-based encryption. In Joachim Biskup and Javier Lopez, editors, *ESORICS*, volume 4734 of *Lecture Notes in Computer Science*, pages 139–154. Springer, 2007.

[AL10] Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, volume 6056 of *Lecture Notes in Computer Science*, pages 384–402. Springer, 2010.

[AMORH14] Gora Adj, Alfred Menezes, Thomaz Oliveira, and Francisco Rodríguez-Henríquez. Computing Discrete Logarithms in $\mathbb{F}_{3^{6*137}}$ and $\mathbb{F}_{3^{6*163}}$ using Magma. Cryptology ePrint Archive, Report 2014/057, 2014. `http://eprint.iacr.org/`.

[Att14] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 557–577. Springer, 2014.

[Att15] Nuttapong Attrapadung. Dual system encryption framework in prime-order groups. *IACR Cryptology ePrint Archive*, 2015:390, 2015.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. Earlier version appeared in the proceedings of CRYPTO 2001.

[BF05] M. Barbosa and P. Farshim. Efficient identity-based key encapsulation to multiple parties. In Nigel P. Smart, editor, *IMA Int. Conf.*, volume 3796 of *Lecture Notes in Computer Science*, pages 428–441. Springer, 2005.

[BGJT14] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2014.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2005.

[BSNS05] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In Serge Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 380–397. Springer, 2005.

[CLL$^+$12] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter IBE and signatures via asymmetric pairings. *IACR Cryptology ePrint Archive*, 2012:224, 2012.

[CM11] Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings – the role of $\psi$ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011.

[CS06a] Sanjit Chatterjee and Palash Sarkar. Generalization of the selective-ID security model for HIBE protocols. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 241–256. Springer, 2006. Revised version available at Cryptology ePrint Archive, Report 2006/203.

[CS06b] Sanjit Chatterjee and Palash Sarkar. Multi-receiver identity-based key encapsulation with shortened ciphertext. In Rana Barua and Tanja Lange, editors, *INDOCRYPT*, volume 4329 of *Lecture Notes in Computer Science*, pages 394–408. Springer, 2006.

[CW13] Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. In Ran Canetti and Juan A. Garay, editors, *CRYPTO (2)*, volume 8043 of *Lecture Notes in Computer Science*, pages 435–460. Springer, 2013. Full version available as IACR Technical Report, 2013/803, `http://eprint.iacr.org/2013/803`.

[Del07] Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*, pages 200–215. Springer, 2007.

[DF02] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In Joan Feigenbaum, editor, *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 61–80. Springer, 2002.

[DF03] Yevgeniy Dodis and Nelly Fazio. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In Yvo Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 100–115. Springer, 2003.

[DPP07] Cécile Delerablée, Pascal Paillier, and David Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Pairing*, volume 4575 of *Lecture Notes in Computer Science*, pages 39–59. Springer, 2007.

[FHH10] Chun-I Fan, Ling-Ying Huang, and Pei-Hsiu Ho. Anonymous multireceiver identity-based encryption. *IEEE Trans. Computers*, 59(9):1239–1249, 2010.

[FN93] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.

[Gal14] Steven Galbraith. New discrete logarithm records, and the death of Type 1 pairings. `http://ellipticnews.wordpress.com/2014/02/01/new-discrete-logarithm-records-and-the-death-of-type-1-pairings/#comment-426`, 2014.

[GKZ14a] Robert Granger, Thorsten Kleinjung, and Jens Zumbrägel. Breaking '128-bit secure' supersingular binary curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$). Cryptology ePrint Archive, Report 2014/119, 2014. `http://eprint.iacr.org/`.

[GKZ14b] Robert Granger, Thorsten Kleinjung, and Jens Zumbragel. Discrete logarithms in $GF(2^9 234)$. `https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1401&L=NMBRTHRY&F=&S=&P=8736`, 2014.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[GR04] Steven D. Galbraith and Victor Rotger. Easy decision-Diffie-Hellman groups. *IACR Cryptology ePrint Archive*, 2004:70, 2004.

[GW09] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.

[Jou13] Antoine Joux. A new index calculus algorithm with complexity $L(1/4+o(1))$ in small characteristic. In Tanja Lange, Kristin Lauter, and Petr Lisonek, editors, *Selected Areas in Cryptography*, volume 8282 of *Lecture Notes in Computer Science*, pages 355–379. Springer, 2013.

[JR13] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2013.

[LOS$^{+}$10] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91. Springer, 2010.

[LSW10] Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation Systems with Very Small Private Keys. In *IEEE Symposium on Security and Privacy*, pages 273–285. IEEE Computer Society, 2010.

[LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 455–479. Springer, 2010.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.

[OT11] Tatsuaki Okamoto and Katsuyuki Takashima. Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption. In Dongdai Lin, Gene Tsudik, and Xiaoyun Wang, editors, *CANS*, volume 7092 of *Lecture Notes in Computer Science*, pages 138–159. Springer, 2011.

[OT12] Tatsuaki Okamoto and Katsuyuki Takashima. Fully Secure Unbounded Inner-Product and Attribute-Based Encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 349–366. Springer, 2012.

[PKL08] Jong Hwan Park, Ki Tak Kim, and Dong Hoon Lee. Cryptanalysis and improvement of a multi-receiver identity-based key encapsulation at INDOCRYPT 06. In Masayuki Abe and Virgil D. Gligor, editors, *ASIACCS*, pages 373–380. ACM, 2008.

[PPS11] Duong Hieu Phan, David Pointcheval, and Mario Strefler. Adaptively secure broadcast encryption with forward secrecy. *IACR Cryptology ePrint Archive*, 2011:463, 2011.

[PPSS12] Duong Hieu Phan, David Pointcheval, Siamak Fayyaz Shahandashti, and Mario Strefler. Adaptive CCA Broadcast Encryption with Constant-Size Secret Keys and Ciphertexts. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *ACISP*, volume 7372 of *Lecture Notes in Computer Science*, pages 308–321. Springer, 2012.

[RNZ14] Yanli Ren, Zihua Niu, and Xinpeng Zhang. Fully Anonymous Identity-Based Broadcast Encryption without Random Oracles. *International Journal of Network Ssecurity*, 16(4):256–264, 2014.

[RS13] Somindu C. Ramanna and Palash Sarkar. (Anonymous) Compact HIBE From Standard Assumptions. *IACR Cryptology ePrint Archive*, 2013:806, 2013.

[Sma04] Nigel P. Smart. Efficient key encapsulation to multiple parties. In Carlo Blundo and Stelvio Cimato, editors, *SCN*, volume 3352 of *Lecture Notes in Computer Science*, pages 208–219. Springer, 2004.

[SV07] Nigel P. Smart and Frederik Vercauteren. On computable isomorphisms in efficient asymmetric pairing-based systems. *Discrete Applied Mathematics*, 155(4):538–547, 2007.

[Ver04] Eric R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. *J. Cryptology*, 17(4):277–296, 2004.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.

[Wat09] Brent Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.

# A Proof of Proposition 2.1

Suppose that $\mathcal{O}_1$ is the identity element of $\mathbb{G}_1$.

To prove the first point, we show that an algorithm $\mathscr{A}^*$ to solve DDH1$^*$ can be used to build an algorithm $\mathscr{A}$ to solve DDH1 such that $\mathscr{A}$ and $\mathscr{A}^*$ require the same time and $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}^*}(\mathscr{A}^*) \leq \mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A})$. The construction of $\mathscr{A}$ is the following:

```
$\mathscr{A}(\mathcal{G}, Q_1, R_1, S_1)$:
  if $Q_1 = \mathcal{O}_1$
    if $S_1 = \mathcal{O}_1$ return 1; else return 0;
  else
    return $\mathscr{A}^*(\mathcal{G}, Q_1, S_1, R_1)$.
```

Clearly, $\mathscr{A}$ takes the same time as $\mathscr{A}^*$. We now consider the advantage of $\mathscr{A}$.

The input to $\mathscr{A}$ is $(\mathcal{G}, Q_1 = aP_1, R_1 = bP_1, S_1 = zP_1)$ where $z$ is either $ab$ or $z = ab + c$ with $c$ being a random element of $\mathbb{Z}_p$. In the later case, $z$ is also uniformly distributed over $\mathbb{Z}_p$. Note that $Q_1 = \mathcal{O}_1$ if and only if $a = 0$. In this case, $z = ab$ if and only if $S_1 = \mathcal{O}_1$. So, if $a = 0$, then $\mathscr{A}$ correctly solves the corresponding DDH1 instance without making a call to $\mathscr{A}^*$.

Let us now consider the case $a \neq 0$, an event which occurs with probability $(p-1)/p$. For any $\alpha \in \mathbb{Z}^\times$, $\Pr[a = \alpha, a \neq 0] = \Pr[a = \alpha]$ and the conditional probability $\Pr[a = \alpha | a \neq 0] = \Pr[a = \alpha, a \neq 0]/\Pr[a \neq 0] = \Pr[a = \alpha]/\Pr[a \neq 0] = 1/(p-1)$, i.e., conditioned on the event $a \neq 0$, $a$ is uniformly distributed over $\mathbb{Z}_p^\times$.

The call to $\mathscr{A}^*$ is on the input $(\mathcal{G}, Q_1, S_1, R_1)$. Let $\mu = b - a^{-1}z = a^{-1}c$ and $s = b - \mu$. If $c = 0$, i.e., $z = ab$, then $\mu = 0$ and $(Q_1, R_1, S_1) = (aP_1, sP_1, asP_1)$. Suppose that $c$ is uniformly distributed over $\mathbb{Z}_p$. In this case, $(Q_1, R_1, S_1) = (aP_1, (s + \mu)P_1, asP_1)$. The following calculations show that conditioned on the event $a \neq 0$, the distribution of $\mu$ is uniform over $\mathbb{Z}_p$ and that $\mu$ and $a$ are conditionally independent. For any $\beta \in \mathbb{Z}_p$,

$$\begin{aligned}
\Pr[\mu = \beta | a \neq 0] &= \Pr[a^{-1}c = \beta | a \neq 0] \\
&= \Pr[c = a\beta | a \neq 0] \\
&= \sum_{\gamma \in \mathbb{Z}_p^\times} \Pr[c = a\beta | a \neq 0, a = \gamma] \times \Pr[a = \gamma | a \neq 0] \\
&= 1/(p-1) \times \sum_{\gamma \in \mathbb{Z}_p^\times} \Pr[c = \gamma\beta | a \neq 0] \\
&= 1/(p-1) \times (p-1)/p \text{ (since c and a are independent)} \\
&= 1/p.
\end{aligned}$$

For $\alpha \in \mathbb{Z}_p^\times$ and $\beta \in \mathbb{Z}_p$,

$$\begin{aligned}
\Pr[\mu = \beta | a = \alpha, a \neq 0] &= \Pr[a^{-1}c = \beta | a = \alpha, a \neq 0] \\
&= \Pr[c = a\beta | a = \alpha, a \neq 0] \\
&= \Pr[c = \alpha\beta | a \neq 0] \\
&= 1/p \text{ (since a and c are independent).}
\end{aligned}$$

The above calculations show that conditioned on the event $a \neq 0$, the call to $\mathscr{A}^*$ on the input $(Q_1, S_1, R_1)$ determines whether $\mu = 0$ (corresponding to $z = ab$) or whether $\mu$ is a random element of $\mathbb{Z}_p$ (corresponding to $c$ being a random element of $\mathbb{Z}_p$) which is independent of $a$. An easier calculation proves that $\mu$ is also independent of $b$.

Let the internal random bits of $\mathscr{A}^*$ be denoted by $r$. Since $\mathscr{A}$ does not use any extra internal randomness, the internal random bits of $\mathscr{A}$ are also $r$. To relate the advantages of $\mathscr{A}$ and $\mathscr{A}^*$ we need to compute some probabilities. The first calculation is for $z = ab$.

$$\begin{aligned}
&\Pr_{a,b,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, abP_1) = 1] \\
&= \Pr_{a,b,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, abP_1) = 1 | a = 0] \times \Pr_a[a = 0] + \Pr_{a,b,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, abP_1) = 1 | a \neq 0] \times \Pr_a[a \neq 0] \\
&= 1/p + (p-1)/p \times \Pr_{a,s,r}[\mathscr{A}^*(\mathcal{G}, aP_1, asP_1, sP_1) = 1].
\end{aligned}$$

Next consider that $z = ab + c$, where $c$ is an independent uniform element of $\mathbb{Z}_p$.

$$\begin{aligned}
&\Pr_{a,b,c,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, (ab+c)P_1) = 1] \\
&= \Pr_{a,b,c,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, (ab+c)P_1) = 1 | a = 0] \times \Pr[a = 0] \\
&\quad + \Pr_{a,b,c,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, (ab+c)P_1) = 1 | a \neq 0] \times \Pr[a \neq 0] \\
&= 1/p \Bigg( \Pr_{a,b,c,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, (ab+c)P_1) = 1 | a = 0, c = 0] \times \Pr[c = 0] \\
&\quad + \Pr_{a,b,c,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, (ab+c)P_1) = 1 | a = 0, c \neq 0] \times \Pr[c \neq 0] \Bigg) \\
&\quad + (p-1)/p \times \Pr_{a,s,\mu,r}[\mathscr{A}^*(\mathcal{G}, aP_1, asP_1, (s+\mu)P_1) = 1] \\
&= 1/p^2 + (p-1)/p \times \Pr_{a,s,\mu,r}[\mathscr{A}^*(\mathcal{G}, aP_1, asP_1, (s+\mu)P_1) = 1].
\end{aligned}$$

The relation between the advantages is obtained as follows.

$$\begin{aligned}
&\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A}) \\
&= \left| \Pr_{a,b,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, abP_1) = 1] - \Pr_{a,b,c,r}[\mathscr{A}(\mathcal{G}, aP_1, bP_1, (ab+c)P_1) = 1] \right| \\
&= (p-1)/p^2 \\
&\quad + (p-1)/p \times \left| \Pr_{a,s,r}[\mathscr{A}^*(\mathcal{G}, aP_1, asP_1, sP_1) = 1] - \Pr_{a,s,\mu,r}[\mathscr{A}^*(\mathcal{G}, aP_1, asP_1, (s+\mu)P_1) = 1] \right| \\
&= (p-1)/p^2 + (p-1)/p \times \mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}^*}(\mathscr{A}^*).
\end{aligned}$$

From this we obtain,

$$\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}^*}(\mathscr{A}^*) \quad = \quad p/(p-1) \times \mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A}) - 1/p.$$

If $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A}) \leq (p-1)/p$, then we get $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}^*}(\mathscr{A}^*) \leq \mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A})$ as required.

Consider the second point of the proposition. This is proved by showing that any algorithm $\mathscr{A}$ to solve DDH1 can be used to build an algorithm $\mathscr{A}^*$ to solve DDH1$^*$ such that $\mathscr{A}$ and $\mathscr{A}^*$ require the same time and $\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A}) = \mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}^*}(\mathscr{A}^*)$. The construction of $\mathscr{A}^*$ is the following:

```
$\mathscr{A}^*(\mathcal{G}, Q_1, R_1, S_1)$:
    return $\mathscr{A}(\mathcal{G}, Q_1, S_1, R_1)$.
```

Clearly, $\mathscr{A}^*$ takes the same time as $\mathscr{A}$. We now consider the advantage of $\mathscr{A}^*$.

The input to $\mathscr{A}^*$ will be a tuple $(\mathcal{G}, aP_1, asP_1, dP_1)$ where $d$ is either $s$ or $s + \mu$ for an independent and random $\mu$. Also, note that by the definition of the DDH1$^*$ problem $a \neq 0$. Let $b = s + \mu$ and $c = -a\mu$ so that $a(b - \mu) = ab - a\mu = ab + c$ and since $a \neq 0$, for a uniform random $\mu$, $c$ is also distributed randomly over $\mathbb{Z}_p$ and is independent of $a$ and $b$. If $\mu = 0$, then $(Q_1, R_1, S_1) = (aP_1, abP_1, bP_1)$ and if $\mu$ is a random element of $\mathbb{Z}_p$, then $(Q_1, R_1, S_1) = (aP_1, (ab + c)P_1, bP_1)$.

$\mathscr{A}^*$ returns 1 on input $(\mathcal{G}, Q_1, R_1, S_1)$ if and only if $\mathscr{A}$ returns 1 on input $(\mathcal{G}, Q_1, S_1, R_1)$. Consequently, the advantages of $\mathscr{A}$ and $\mathscr{A}^*$ are equal. □