Efficient Adaptively Secure IBBE from Standard Assumptions

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

May 28, 2014

Abstract

This paper describes the first construction of efficient identity-based broadcast encryption (IBBE) schemes which can be proved secure against adaptive-identity attacks based on standard assumptions. The constructions are obtained by extending the currently known most efficient identity-based encryption scheme proposed by Jutla and Roy in 2013. Ciphertext size and user storage compare favourably to previously known constructions. The new constructions fill both a practical and a theoretical gap in the literature on efficient IBBE schemes.
Keywords: identity-based encryption, broadcast encryption, identity-based broadcast encryption, Type-3 pairings, dual-system technique.

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 and anybody can broadcast an encrypted a message which can be decrypted by the intended set of privileged users.

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. 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 (summarised sID), requires the adversary to specify the target set *before* it can corrupt any entity. The stronger notion, called adaptive-identity security (summarised 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.

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. 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 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 preferrable.

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: Constructions of IBBE falls in the general category of pairing-based cryptography. Such constructions require a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of some prime order $\mathbb{Z}_p$. 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 [BGJT13, 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 efficient IBBE constructions that achieve security against adaptive-identity attacks under standard hardness assumptions. All previously known schemes either achieved security against selective-identity attacks, or used non-standard assumptions, or could be obtained by specialising inner product encryption making them quite inefficient. Further, our constructions are based on Type-3 pairings, whereas previous works on IBBE used Type-1 pairings.

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. An IBE encryption results in a ciphertext which consist of several elements of $\mathbb{G}_1$. 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] (actually a simple variant of their original proposal is the IBE of choice). In this work, we investigate the possibility of converting this IBE scheme into an IBBE scheme. The intuitive idea is to use 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 assumptions consists of the standard DDH assumption in the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ (DDH1 and DDH2 respectively).

Ciphertexts in $\mathcal{IBBE}_1$ contains ℓ $\mathbb{Z}_p$ elements (called tags) where ℓ 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 describe a method 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 substree (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 $\mathcal{IBBE}_1$.

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. Delerablee, Paillier and Pointcheval introduced dynamic broadcast encryption in [DPP07] and proposed two (partially) adaptively secure constructions.

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 consists of constant-size ciphertexts while the former does not.

All the schemes mentioned so far are secure under some non-standard and parametrised assumptions. The first BE scheme secure proved secure under static assumptions was proposed by Waters [Wat09] using the dual system encyrption 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, Farshim [BF05] and independently by Baek, Safavi-Naini, 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 is m, the maximum number of recipients of a ciphertext. Both schemes ([PKL08] and [AKN07]) are selectively secure without random oracles. In 2007, Delerablee [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. 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 [LOS+10, OT11, OT12]. Most adaptively secure inner product encryption schemes are constructed using dual pairing vector spaces and hence lead to very inefficient schemes. This is because ciphertexts and keys usually consists vectors of dimension at least 4 (over $\mathbb{G}_1$ or $\mathbb{G}_2$) resulting in too much ciphertext overhead in the broadcast setting. It is due to this reason, we do not compare with IBBE schemes that can be obtained by specialising inner product encryption.

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 repectively. 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 Delerablee [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 refered 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 elements in Hdr (#hdr) from $\mathbb{G}_1$ and $\{0,1\}^n$ (in case a KDF is used), number of elements in a user key (#ukey) from $\mathbb{G}_2$ and $\mathbb{Z}_p$, number of pairings required for decryption, security model and computational assumptions. m denotes the maximum size of the privileged users' set. ℓ ($\leq m$) 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 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 ℓ_1 separate symmetric encryptions of the message under the ℓ_1 keys generated by calls to the encapsulation algorithm of construction (a). In practice, the ℓ_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.

The short-hand 'sID' is used to indicate selective identity security and 'aID' is used to indicate security against adaptive-identity attacks. '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 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.

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), security of a pseudorandom function (PRF) and external DH (XDH).

The XDH 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		#dec	Security	Assumptions
		$\mathbb{G}_1$	$\mathbb{G}_T$	$\mathbb{G}_1$	$\{0,1\}^\kappa$	$\mathbb{G}_2$	$\mathbb{Z}_p$			
[BSNS05]	Type-1	3	–	$\ell+1$	–	1	–	2	sID-CPA	DBDH
[BF05]	Type-1	3	–	3ℓ	–	1	–	2	sID-CCA	Gap-BDH
[AKN07] (from GS-HIBE)	Type-1	$m+2$	1	$\ell+1$	–	$O(m)$	–	$\ell+1$	sID-CPA	DBDH
[Del07]-ROM	Type-1	$m+2$	1	2	–	1	1	2	sID-CPA	GDDHE
[GW09]-(a)-ROM	Type-1	$4m+2$	–	4	–	1	1	2	aID-CPA	m-DBDHES
$\mathcal{IBBE}_1^{\rm RO}$	Type-3	$m+4$	1	$\ell+2$	1	5	–	3	aID-CPA	XDH
$\mathcal{IBBE}_2^{\rm RO}$	Type-3	m_2+4	1	$\ell+2\ell_1$	ℓ_1+1	5	–	3	aID-CPA	XDH

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

Scheme	Pairing	#pp		#hdr			#ukey		#dec	Security	Assumptions
		$\mathbb{G}_1$	$\mathbb{G}_T$	$\mathbb{G}_1$	$\{0,1\}^\kappa$	$\mathbb{Z}_p$	$\mathbb{G}_2$	$\mathbb{Z}_p$			
[CS06a]	Type-1	$m+4$	–	$\ell+1$	–	–	2	–	2	sID-CPA	DBDH
[CS06b]	Type-1	$m+4$	–	2ℓ	–	–	$m+2$	–	2	sID-CPA	$(m+1)$-DBDHE
[AKN07] (from BBG-HIBE)	Type-1	$m+4$	–	2	–	–	$\ell+1$	–	2	sID-CPA	$(\ell-1)$-DBDHE
[Del07]	Type-1	$m+2$	1	2	–	–	1	–	2	sID-CPA	GDDHE
[GW09]-(a)	Type-1	$4m+2$	–	4	–	ℓ	1	1	2	aID-CPA	m-DBDHES, PRF
[GW09]-(b)	Type-1	$4m_2+2$	–	$4\ell_1$	ℓ_1	ℓ_2	1	1	2	aID-CPA	m-DBDHES, PRF
$\mathcal{IBBE}_1$	Type-3	$m+4$	1	$\ell+2$	–	ℓ	5	–	3	aID-CPA	XDH
$\mathcal{IBBE}_2$	Type-3	m_2+4	1	$\ell+2\ell_1$	ℓ_1	m_2	5	–	3	aID-CPA	XDH

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 $\mathbb{G}_1$.

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

1. Apart from $\mathcal{IBBE}_1$, $\mathcal{IBBE}_1^{\rm RO}$, $\mathcal{IBBE}_2$, $\mathcal{IBBE}_2^{\rm RO}$ and the constructions of Gentry and Waters [GW09] (denoted (a), (b), (a)-ROM), all other schemes listed in the tables are secure only in the weaker selective identity model.

2. $\mathcal{IBBE}_1$, $\mathcal{IBBE}_2$ are the only known constructions to achieve adaptive security from the standard and static DDH assumptions. The GW constructions are based on non-standard assumptions.

3. The GW constructions 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.

4. Even though the new constructions achieve stronger security from the standard XDH assumption, this is not done at a loss in efficiency. The ciphertext size, user storage and also the encryption and decryption times are comparable to previous constructions. Apart from the comparison of the number of group elements, it is also to be noted that the new constructions use Type-3 pairings whereas the previous constructions used Type-1 pairings. This leads to significantly smaller sizes for $\mathbb{G}_1$ which leads to smaller ciphertexts and faster encryption and decryption algorithms.

2 Preliminaries

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

2.1 Notation

The notation $x_1, \ldots, x_k \xleftarrow{\text{R}} \mathcal{X}$ 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. For a (probabilistic) algorithm $\mathcal{A}$, $y \xleftarrow{\text{R}} \mathcal{A}(x)$ means that y is chosen according to the output distribution of $\mathcal{A}$ on input x. $\mathcal{A}(x; r)$ 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 $\mathbb{G}$.

2.2 Asymmetric Pairings and Hardness Assumptions

A bilinear pairing 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 assumptions based on which the security of our constructions is proven are the decision Diffie-Hellman (DDH) assumptions in groups $\mathbb{G}_1$ and $\mathbb{G}_2$, called DDH1 and DDH2 respectively. Below, we describe these two assumptions. Let $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ be an asymmetric pairing and $\mathscr{A}$, a probabilistic polynomial time (PPT) algorithm $\mathscr{A}$ that outputs 0 or 1.

Assumption DDH1. Define a distribution $\mathcal{D}$ as follows: $P_1 \xleftarrow{\text{U}} \mathbb{G}_1^\times$; $P_2 \xleftarrow{\text{U}} \mathbb{G}_2^\times$, $a, s \xleftarrow{\text{U}} \mathbb{Z}_p$, $\mu \xleftarrow{\text{U}} \mathbb{Z}_p$; $\mathcal{D} = (\mathcal{G}, P_1, aP_1, asP_1)$. The advantage of $\mathscr{A}$ in solving the DDH1 problem is given by

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

Essentially, $\mathscr{A}$ has to decide whether $\mu = 0$ or $\mu \in_{\text{U}} \mathbb{Z}_p$ given $(\mathcal{D}, (s + \mu)P_1)$. The (ε, t)-DDH1 assumption holds in $\mathcal{G}$ if for any adversary $\mathscr{A}$ running in time at most t, $\mathsf{Adv}_{\mathcal{G}}^{\text{DDH1}}(\mathscr{A}) \leq \varepsilon$.

Assumption DDH2. Let a distribution $\mathcal{D}$ be defined as follows: $P_1 \xleftarrow{\text{U}} \mathbb{G}_1^\times$; $P_2 \xleftarrow{\text{U}} \mathbb{G}_2^\times$, $r, c \xleftarrow{\text{U}} \mathbb{Z}_p$, $\gamma \xleftarrow{\text{U}} \mathbb{Z}_p$;

$$\mathcal{D} = (\mathcal{G}, P_1, P_2, rP_2, cP_2).$$

$\mathscr{A}$'s advantage in solving the DDH2 problem is given by

$$\mathsf{Adv}_{\mathcal{G}}^{\text{DDH2}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{D}, rcP_2) = 1] - \Pr[\mathscr{A}(\mathcal{D}, (rc+\gamma)P_2) = 1]|.$$

The (ε, t)-DDH2 assumption is that, for any t-time algorithm $\mathscr{A}$, $\mathsf{Adv}_{\mathcal{G}}^{\text{DDH2}}(\mathscr{A}) \leq \varepsilon$.

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 uses to encrypt the message via the data encapsulation mechanism (DEM). The DEM can be instantiated to any 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 symmetric encryption portion (DEM) is omitted and also not considered in the security proof. 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 – Setup, Encap, KeyGen and Decap. The identity space is denoted $\mathscr{I}$ and the key space for the symmetric encryption scheme is denoted by $\mathscr{K}$.

$\mathsf{Setup}(\kappa, m)$ Takes as input a security parameter κ and the maximum number m of identities in a privileged recipient group. It outputs the public parameters $\mathcal{PP}$ and the master secret $\mathcal{MSK}$.

$\mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id})$ Input is an identity id and master secret $\mathcal{MSK}$; output is a secret key $\mathcal{SK}_{\mathsf{id}}$ for 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 (Hdr, K) where 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 id, a secret key $\mathcal{SK}_{\mathsf{id}}$ corresponding to id_i, a header Hdr and outputs the session key K if $\mathsf{id} \in S$.

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

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 $(\mathcal{PP}, \mathcal{MSK}) \xleftarrow{\text{R}} \mathsf{Setup}(\kappa, \mathscr{I}, m)$, $\mathcal{SK}_{\mathsf{id}_i} \xleftarrow{\text{R}} \mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id}_i)$, $(\mathsf{Hdr}, K) \xleftarrow{\text{R}} \mathsf{Encap}(\mathcal{PP}, S)$, the $\Pr[K = \mathsf{Decap}(\mathcal{PP}, S, \mathsf{id}_i, \mathcal{SK}_{\mathsf{id}_i}, \mathsf{Hdr})] = 1$.

Definition 2.2 (IBBE Security)**.** Adaptive security against chosen plaintext attacks in identity-based broadcast encryption systems is defined via the following game ind-cpa between an adversary $\mathscr{A}$ and a challenger.

Setup: The challenger runs the 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 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 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 β 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 β'.

If $\beta = \beta'$, then $\mathscr{A}$ wins the game. The advantage of $\mathscr{A}$ of the IBBE scheme in winning the $\mathsf{ind\text{-}cpa}$ 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 (ε, 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 vairiant of the Jutla-Roy IBE [JR13] defined in [RS13] refered to as JR-IBE-D. The following section describes the IBE scheme.

3.1 Variant of Jutla-Roy IBE

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 $P_1 \xleftarrow{\text{U}} \mathbb{G}_1^\times$, $P_2 \xleftarrow{\text{U}} \mathbb{G}_2^\times$, $\alpha_1, \alpha_2, \Delta_1, \Delta_2, \Delta_3, c, d, e \xleftarrow{\text{U}} \mathbb{Z}_p$, $b \xleftarrow{\text{U}} \mathbb{Z}_p^\times$, and set $U_1 = (-\Delta_1 b + d)P_1$, $V_1 = (-\Delta_2 b + e)P_1$, $W_1 = (-\Delta_3 b + c)P_1$, $g_T = e(P_1, P_2)^{\alpha_1 + b\alpha_2}$. The parameters are given by

$\mathcal{PP} : (P_1, bP_1, U_1, V_1, W_1, g_T)$
$\mathcal{MSK} : (P_2, cP_2, \alpha_1, \alpha_2, \Delta_1, \Delta_2, \Delta_3, d, e)$

Ciphertext:

$\mathsf{tag}, s \xleftarrow{\text{U}} \mathbb{Z}_p$, $\{\mu \xleftarrow{\text{U}} \mathbb{Z}_p\}$
$C_0 = m \cdot (g_T)^s \{\times e(P_1, P_2)^{u\mu}\}$,
$C_1 = sP_1\{+\mu P_1\}$, $C_2 = sbP_1$, $C_3 = s(U_1 + \mathsf{id}V_1 + \mathsf{tag}W_1)\{+\mu(d + \mathsf{id} \cdot e + \mathsf{tag} \cdot c)P_1\}$.

Key:

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

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 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 id of its choice. This header when decapsulated using a secret key for 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 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 sovle the particular instance. At the same time the simulator needs to create a valid semi-functional ciphertext for the challenge identity $\widehat{\mathsf{id}}$. One must also ensure any semi-functional ciphertext that the simulator creates for 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 π 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 id' is by setting $\mathsf{tag}' = f(\mathsf{id}')$ implying that any attempt by the simulator to create a semi-functional ciphertext for id will set $\mathsf{tag} = \pi$. As a result, decryption is successful and the simulator gains no information about 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 π 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 π, 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 id will now consist of the two subhashes $\sum_{j=0}^{m}(\mathsf{id})^j e_j$ 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 $\mathcal{IBBE}_1$

We define our first IBBE construction

$$\mathcal{IBBE}_1 = (\mathcal{IBBE}_1.\mathsf{Setup}, \mathcal{IBBE}_1.\mathsf{Encrypt}, \mathcal{IBBE}_1.\mathsf{KeyGen}, \mathcal{IBBE}_1.\mathsf{Decrypt})$$

as follows.

$\mathcal{IBBE}_1.\mathsf{Setup}(\kappa, m)$: Generate a Type-3 pairing $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, F_1, F_2)$ based on the security parameter κ. Here, $\mathscr{I} = \mathbb{Z}_p$ and $\mathscr{K} = \mathbb{G}_T$. In practice, a hash function can be used to map $\mathbb{G}_T$ to the actual key space of the symmetric encryption scheme. Compute parameters as follows.

$$P_1 \xleftarrow{\text{U}} \mathbb{G}_1^\times,\ P_2 \xleftarrow{\text{U}} \mathbb{G}_2^\times$$
$$\alpha_1, \alpha_2, c, \Delta, (e_j, \Delta_j)_{j=0}^m \xleftarrow{\text{U}} \mathbb{Z}_p,\ b \xleftarrow{\text{U}} \mathbb{Z}_p^\times,$$

$$U_{1,j} = (\Delta_j b + e_j)P_1 \text{ for } j = 0, \ldots, m,\ W_1 = (\Delta b + c)P_1,$$
$$g_T = e(P_1, P_2)^{\alpha_1 b + \alpha_2},$$

$$\mathcal{PP} : (P_1, bP_1, (U_{1,j})_{j=0}^m, W_1, g_T)$$
$$\mathcal{MSK} : (P_2, cP_2, \alpha_1, \alpha_2, \Delta, (e_j, \Delta_j)_{j=0}^m)$$

$\mathcal{IBBE}_1$.KeyGen($\mathcal{MSK}$, id): Choose $r \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ and compute the secret key $\mathcal{SK}_{\mathsf{id}} = (D_1, D_2, D_3, D_4, D_5)$ as follows.

$$D_1 = rP_2,\ D_2 = rcP_2,\ D_3 = \left(\alpha_1 + r(\textstyle\sum_{j=0}^{m}(\mathsf{id})^j e_j)\right) P_2,$$
$$D_4 = r\Delta P_2,\ D_5 = \left(\alpha_2 + r(\textstyle\sum_{j=0}^{m}(\mathsf{id})^j \Delta_j)\right) P_2,$$

$\mathcal{IBBE}_1$.Encap($\mathcal{PP}, S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\}$): If $\ell \le m$, pick $s, (\mathsf{tag}_i)_{i=1}^{\ell} \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Compute the session key as $K = g_T^s$. The header is given by $\mathsf{Hdr} = (C_1, C_2, (C_{3,i}, \mathsf{tag}_i)_{i=1}^{\ell})$ where

$$C_1 = sP_1,\ C_2 = sbP_1,$$
$$C_{3,i} = s(\textstyle\sum_{j=0}^{m}(\mathsf{id}_i)^j U_{1,j} + \mathsf{tag}_i W_1) \text{ for } i = 1, \ldots, \ell.$$

$\mathcal{IBBE}_1$.Decap($\mathcal{PP}, S, \mathsf{id}, \mathcal{SK}_{\mathsf{id}}, \mathsf{Hdr}$): Suppose that $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\}$. If $\mathsf{id} \in S$, there is some index $i \in [1, \ell]$ such that $\mathsf{id} = \mathsf{id}_i$. The session key is derived as follows.

$$K = \frac{e(C_1, \mathsf{tag}_i D_2 + D_3) e(C_2, \mathsf{tag}_i D_4 + D_5)}{e(C_{3,i}, D_1)}.$$

Correctness: Let $S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\} \subseteq \mathscr{I}$ with $\ell \le m$. 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 $\mathcal{SK}_{\mathsf{id}_i} \xleftarrow{\mathrm{R}} \mathcal{IBBE}_1.\mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id}_i; r)$ 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 $\mathbb{G}_1$, ℓ 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 $\mathbb{G}_2$.

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 κ, an index $i \in [1, m]$ as input and produces a value in $\mathbb{Z}_p$ as output. If H is modeled 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 κ-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

$\mathcal{IBBE}_1^{\text{RO}}$ 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 π 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 ℓ 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 $\ell \leq m$.

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

Proof. 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.

$\mathcal{IBBE}_1.\mathsf{SFEncap}(\mathcal{PP}, \mathcal{MSK}, S, (\mathsf{Hdr}, K))$: Takes as input a header-key pair 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.

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

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

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 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 (Hdr, K) is normal: Let $\mathcal{SK}_{\mathsf{id}_i} \longleftarrow \mathcal{IBBE}_1.\mathsf{SFKeyGen}(\mathcal{MSK}, \mathcal{SK}'_{\mathsf{id}_i}; \gamma, \pi)$ where $\mathcal{SK}'_{id_i}$ is a normally generated key for id_i. The requirement is that when 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 (Hdr, K) is semi-functional: Let (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 (Hdr, K) are semi-functional: Let (Hdr', K') be a normally generated header-key pair and $\mathcal{SK}_{\mathsf{id}_i}$ a normal key for id_i. Also let $(\mathsf{Hdr}, K) \longleftarrow \mathcal{IBBE}_1.\mathsf{SFEncap}(\mathcal{PP}, \mathcal{MSK}, S, (\mathsf{Hdr}', K'); \mu)$ and $\mathcal{SK}_{\mathsf{id}_i} \longleftarrow \mathcal{IBBE}_1.\mathsf{SFKeyGen}(\mathcal{MSK}, \mathcal{SK}'_{\mathsf{id}_i}; \gamma, \pi)$. 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 Hdr are called *nominally semi-functional*.

Now, given that semi-functional algorithms are defined, consider a sequence of games G_{real}, G_0, $(\mathsf{G}_k)_{k=1}^{q}$, G_{final} between an adversary $\mathscr{A}$ and a challenger with the games defined as follows.

- G_{real}: the actual IBBE security game $\mathsf{ind\text{-}cpa}$ (described in Section 2.3).
- G_k, $0 \leq k \leq q$: challenge header is semi-functional; K_0 is semi-functional; first k user keys are semi-functional.
- G_{final}: challenge header is semi-functional; K_0 is random.

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

- $|\Pr[X_{real}] - \Pr[X_0]| \leq \varepsilon_{\text{DDH1}}$,
- $|\Pr[X_{k-1}] - \Pr[X_k]| \leq \varepsilon_{\text{DDH2}}$,
- $|\Pr[X_q] - \Pr[X_{final}]| \leq q/p$.

Clearly, the bit β is statistically hidden from the attacker in G_{final}, which means that $\Pr[X_{final}] = 1/2$. Hence, the advantage of $\mathscr{A}$ in breaking the security of $\mathcal{IBBE}_1$ is thus given by

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

□

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 G_{real} and G_0, can solve DDH1. Similarly, $\mathscr{A}$'s power to distinguish between G_{k-1} and G_k for $k \in [1, q]$, can be leveraged to build a DDH2-solver $\mathscr{B}_2$.

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

Proof. Let $(\mathcal{G}, P_1, bP_1, sbP_1, P_2, (s+\mu)P_1)$ be the instance of DDH1 that $\mathscr{B}_1$ has to solve i.e., decide whether $\mu = 0$ or $\mu \in_{\text{U}} \mathbb{Z}_p$. 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:

$U_{1,j} = \Delta_j(bP_1) + e_jP_1$ 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 $\mathcal{IBBE}_1$.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 $\widehat{\mathsf{tag}}_i \xleftarrow{\text{U}} \mathbb{Z}_p$,
$K_0 = e(sbP_1, P_2)^{\alpha_2} e((s+\mu)P_1, P_2)^{\alpha_1} = g_T^s e(P_1, P_2)^{\alpha_1 \mu}$,
$\widehat{C}_1 = (s + \mu)P_1 = sP_1 + \mu P_1$,
$\widehat{C}_2 = sbP_1$,

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

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

$\mathscr{B}_1$ 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 β' 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 $\mathsf{Adv}_{\mathcal{G}}^{\text{DDH1}}(\mathscr{B}_1) \leq \varepsilon_{\text{DDH1}}$, 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 $(\mathcal{G}, P_1, P_2, rP_2, cP_2, (rc + \gamma)P_2)$ 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 $\alpha_1, \alpha_2', c, \Delta', (e_{j,1}, e_{j,2}, \Delta_j')_{j=0}^m \xleftarrow{\text{U}} \mathbb{Z}_p$ 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 $j = 0, \ldots, m$, $W_1 = -\Delta' P_1$,
$g_T = e(P_1, P_2)^{\alpha'_2}$
$\mathcal{PP} : (P_1, bP_1, (U_{1,j})_{j=0}^m, W_1, g_T)$

The elements Δ, Δ_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 ν-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 $\mathcal{IBBE}_1$.KeyGen needs a modification. The components of the key are computed as shown below.

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

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

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

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

$$\begin{aligned} D_5 &= b^{-1}\left(\alpha'_2 - \alpha_1 + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j (\Delta'_j - e_{j,1})\right)\right) P_2 - b^{-1} r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j e_{j,2}\right)(cP_2) \\ &= b^{-1}\left(\alpha'_2 - \alpha_1 + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j (\Delta'_j - e_{j,1} - c e_{j,2})\right)\right) P_2 \\ &= \left(\frac{\alpha'_2 - \alpha_1}{b} + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j \left(\frac{\Delta'_j - e_j}{b}\right)\right)\right) P_2 \\ &= \left(\alpha_2 + r_\nu \left(\sum_{j=0}^{m} (\mathsf{id}_\nu)^j \Delta_j\right)\right) P_2. \end{aligned}$$

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 $P_2, cP_2, rP_2, (rc+\gamma)P_2$) in the key $\mathcal{SK}_{\mathsf{id}_k} = (D_1, \ldots, D_5)$ for id_k by generating the components as shown below.

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

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

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

$$\begin{aligned}
D_5 &= b^{-1}\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j(\Delta'_j - e_{j,1})\right)(rP_2) - b^{-1}\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j e_{j,2}\right)(rc+\gamma)P_2 \\
&= b^{-1}r\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j(\Delta'_j - e_j)\right)P_2 - b^{-1}\gamma\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j e_{j,2}\right)P_2 \\
&= r\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j\left(\frac{\Delta'_j - e_j}{b}\right)\right)P_2 - \left(\frac{\gamma}{b}\right)\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j e_{j,2}\right)P_2 \\
&= r\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j \Delta_j\right)P_2 - \left(\frac{\gamma}{b}\right)\left(\sum_{j=0}^{m}(\mathsf{id}_k)^j e_{j,2}\right)P_2,
\end{aligned}$$

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 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 β' of β.

$\mathscr{B}_2$ outputs 1 if $\mathscr{A}$ wins and 0 otherwise. Also, $\mathscr{B}_2$ simulates G_{k-1} if $\gamma = 0$ and 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, γ from the instance are uniformly and independently distributed in their respective domains. These scalars determine the distribution of the following quantities.

- α_2, Δ
- $(e_j)_{j=0}^m$ and hence $(\Delta_j)_{j=0}^m$
- r_k, γ_k
- π_k
- $\widehat{\mathsf{tag}}_1, \ldots, \widehat{\mathsf{tag}}_{\widehat{\ell}}$

(α_2, Δ) are uniquely determined by (α_2', Δ'). Scalars r_k, γ_k have the correct distribution since they are set to r, γ respectively. Also, all other information is independent of r, γ. We will now argue that π_k and $\widehat{\mathsf{tag}}_1, \ldots, \widehat{\mathsf{tag}}_{\widehat{\ell}}$ 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.

- $\mathsf{id}_k, \widehat{\mathsf{id}}_1, \ldots, \widehat{\mathsf{id}}_{\widehat{\ell}}$ 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 $\widehat{\ell}+1$.
- $e_{0,2}, e_{1,2}, \ldots, e_{m,2}$ are information theoretically hidden from $\mathscr{A}$ and also chosen uniformly and independently over $\mathbb{Z}_p$.

From these observations, it follows that $\pi_k, \widehat{\mathsf{tag}}_1, \ldots, \widehat{\mathsf{tag}}_{\widehat{\ell}}$ 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 π_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. $|\Pr[X_q] - \Pr[X_{final}]| \leq q/p$.

Proof. In 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 G_{final} and then argue that the resulting game is indistinguishable from 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 $j = 0, \ldots, m$, $W_1 = \Delta' P_1$,
$g_T = e(P_1, P_2)^{\alpha'_2}$
$\mathcal{PP} : (P_1, bP_1, (U_{1,j})_{j=0}^m, W_1, g_T)$

setting

$$\alpha_2 = \frac{\alpha'_2 - \alpha_1}{b}, \quad \Delta = \frac{\Delta' - c}{b},$$
$$\Delta_j = \frac{\Delta'_j - e_j}{b} \qquad \text{for } j = 0, \ldots, m.$$

Although α_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 α'_2 which is chosen independent of α_1.

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

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

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

$$D_5 = \left(\frac{\alpha'_2 - \pi'}{b}\right) + r\left(\sum_{j=0}^m (\mathsf{id})^j \Delta_j\right) P_2.$$

Computing D_3 as $D_3 = \pi' P_2 + r\left(\sum_{j=0}^m (\mathsf{id})^j e_j\right) P_2$ sets $\pi' = \alpha_1 + \gamma\pi$, where $\gamma\pi$ defines the semi-functional component. The other component where π' is used is D_5 that also fixes $\gamma\pi$ in its semi-functional term. It is necessary to ensure that these two are equal. We show below that D_5 is indeed well-formed in this sense.

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

The scalar π is determined by α_1, π' and γ. It will be uniformly distributed in $\mathbb{Z}_p$ unless $\gamma = 0$. Furthermore, D_3 and D_5 are generated using π' which is chosen independent of α_1, thus making the key independent of α_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 $i = 1, \ldots, \widehat{\ell}$,
$\widehat{C}_{3,i} = s\left(\sum_{j=0}^{m} (\widehat{\mathsf{id}}_i)^j \Delta_j' + \widehat{\mathsf{tag}}_i \Delta'\right) P_1 + \mu\left(d + \sum_{j=0}^{m} (\widehat{\mathsf{id}}_i)^j e_j + \widehat{\mathsf{tag}}_i \cdot c\right) P_1.$

The computation above shows that the challenge header consisting of $\widehat{C}_1, \widehat{C}_2, (\widehat{C}_{3,i})_{i=1}^{\widehat{\ell}}$ is generated independent of α_1. Recall that α_1 is chosen independently and uniformly at random from $\mathbb{Z}_p$. Also, public parameters and all the keys are generated independent of α_1. Hence the conditional distribution of α_1 given the public parameters, keys and the challenge header is the same as its unconditional distribution. As a result, K_0 would be uniformly distributed in $\mathbb{G}_T$ and independent of all other information provided to $\mathscr{A}$. Therefore the bit β is information theoretically hidden from the adversary implying that the resulting game (obtained by modifying $\mathcal{IBBE}_1$.SFKeyGen) is G_{final}.

Suppose the adversary makes queries on $\mathsf{id}_1, \ldots, \mathsf{id}_q$. Let γ_i denote the scalar used in generating the semi-functional components in $\mathcal{SK}_{\mathsf{id}_i}$ and let F_i ($i \in [1, q]$) denote the event that $\gamma_i = 0$. Clearly, $\Pr[F_i] = 1/p$ for a fixed i. Observe that G_q and G_{final} proceed identically unless the failure event $\mathsf{F} = \cup_{i=1}^{q} \mathsf{F}_i$ occurs. By the difference lemma (Shoup [Sho04]), we have $|\Pr[\mathsf{G}_q] - \Pr[\mathsf{G}_{final}]| \leq \Pr[F] \leq \sum_{i=1}^{q} \Pr[\mathsf{F}_i] = q/p$. □

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 ℓ ($\leq m$). 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 Setup algorithm. Suppose we want to encrypt to a set S of size $\ell \leq m$.

1. Express ℓ 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 ℓ_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 ℓ_1 $\mathcal{IBBE}_1$ headers (refered 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 ℓ_1 sub-headers in addition to ℓ_1 elements of $\mathbb{G}_T$ each masking the session key.

The above idea is made concrete below as the scheme

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

whose individual algorithms are as follows.

$\mathcal{IBBE}_2.\mathsf{Setup}(\kappa, m = m_1 m_2 - 1)$: Let $(\mathcal{PP}', \mathcal{MSK}') \xleftarrow{\text{R}} \mathcal{IBBE}_1.\mathsf{Setup}(\kappa, m_2)$. Define $\mathcal{PP} = (\mathcal{PP}', m_2)$ and $\mathcal{MSK} = \mathcal{MSK}'$.

$\mathcal{IBBE}_2.\mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id})$: Return $(\mathcal{SK}_{\mathsf{id}} = (D_1, D_2, D_3, D_4, D_5)) \xleftarrow{\text{R}} \mathcal{IBBE}_1.\mathsf{KeyGen}(\mathcal{MSK}, \mathsf{id})$.

$\mathcal{IBBE}_2.\mathsf{Encap}(\mathcal{PP}, S = \{\mathsf{id}_1, \ldots, \mathsf{id}_\ell\})$: Suppose $\ell = (\ell_1 - 1)m_2 + \ell_2$ with $1 \leq \ell_1 \leq m_1$ and $1 \leq \ell_2 \leq m_2$. Partition the set S into ℓ_1 disjoint subsets $S_1, S_2, \ldots, S_{\ell_1}$ where $|S_j| = m_2$ for all $j \in [1, \ell_1 - 1]$ and $|S_{\ell_1}| = \ell_2$. Choose $(s_j)_{j=1}^{\ell_1}, (\mathsf{tag}_i)_{i=1}^{m_2} \xleftarrow{\text{U}} \mathbb{Z}_p$ and set

$$(\mathsf{Hdr}_t, K_t) \longleftarrow \mathcal{IBBE}_1.\mathsf{Encap}(\mathcal{PP}', S_j; s_j, \mathsf{tag}_1, \ldots, \mathsf{tag}_{m_2}) \text{ for } j = 1, \ldots, \ell_1.$$

Recall that the notation $\mathcal{A}(\cdot; R)$ denotes running the probabilistic algorithm $\mathcal{A}(\cdot)$ with its random bits set to R.

Choose a session key $K' \xleftarrow{\text{U}} \mathbb{G}_T$ and mask it separately using $K_1, \ldots, K_{\ell_1}$ as follows.

$$C_{0,j} = K' \cdot K_j \text{ for } j \in [1, \ell_1]. \tag{2}$$

The header is

$$\vec{\mathsf{Hdr}} = (\mathsf{Hdr}_1, \ldots, \mathsf{Hdr}_{\ell_1}, C_{0,1}, \ldots, C_{0,\ell_1}, \mathsf{tag}_1, \ldots, \mathsf{tag}_{m_2}). \tag{3}$$

The actual message is encrypted using the session key K'.

$\mathcal{IBBE}_2.\mathsf{Decap}(\mathcal{PP}, S, \mathsf{id}, \mathcal{SK}_{\mathsf{id}}, \vec{\mathsf{Hdr}})$: Parse S as $(S_1, \ldots, S_{\ell_1})$ and suppose that $\mathsf{id} \in S_j$ for some $j \in [1, \ell_1]$. The session key K is derived as: $K' = C_{0,t} \cdot K_j^{-1}$ where $K_j = \mathcal{IBBE}_1.\mathsf{Decap}(\mathcal{PP}', S_j, \mathsf{id}, \mathcal{SK}_{\mathsf{id}}, \mathsf{Hdr}_j, (\mathsf{tag}_i)_{i=1}^{m_2})$.

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

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 symmetric encryption scheme and hence will be a κ-bit string, where κ is the security parameter. In this case, the quantities $C_{0,j}$ will be generated as $\mathsf{KDF}(K_j) \oplus K'$, where KDF is a key derivation function which maps an element of $\mathbb{G}_T$ to a κ-bit string. As a result, $C_{0,1}, \ldots, C_{0,\ell_1}$ consists of ℓ_1 κ-bit strings. While considering the efficiency of $\mathcal{IBBE}_2$, we will consider the $C_{0,j}$'s to be κ-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 κ-bit strings will require considering the security of 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 κ-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 ℓ, the header sizes of the two IBBE schemes are comparable. For ℓ 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 ℓ_1 κ-bit masked versions of the session key and a single κ-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 κ-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 ℓ 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 $\mathcal{IBBE}_2$

We show that $\mathcal{IBBE}_2$ is secure if $\mathcal{IBBE}_1$ is secure. More precisely, we prove the following.

Theorem 4.1. *If $\mathcal{IBBE}_1$ is (ε, t, q)-IND-ID-CPA-secure then $\mathcal{IBBE}_2$ is (ε', t', q)-IND-ID-CPA-secure where $\varepsilon' \leq 2m_1\varepsilon$ 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 $\widehat{\ell_1} \leq m_1$ is the size of the partition of the challenge set) all running in time t against $\mathcal{IBBE}_1$ such that $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{IBBE}_2}(\mathscr{A}) \leq \sum_{\nu=1}^{\widehat{\ell_1}} \mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{IBBE}_1}(\mathscr{B}_t)$. Since $\widehat{\ell_1} \leq m_1$, the statement of the theorem follows.

Define the following game sequence: $\mathsf{G}_0, \mathsf{G}_1, \ldots, \mathsf{G}_{\widehat{\ell_1}}$ where G_0 is the real ind-cpa game; in G_ν ($\nu \in [1, \widehat{\ell_1}]$), the first ν 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 G_ν 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 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 $\delta \xleftarrow{\text{U}} \{0,1\}$ and sets

$$(\mathsf{Hdr}_j, K_j) \xleftarrow{\text{R}} \mathcal{IBBE}_1.\mathsf{Encap}(\mathcal{PP}', S_j; (\mathsf{tag}_i)_{i=1}^{\widehat{m_2}}), \text{ for } j \in [1, \widehat{\ell_1}] \setminus \{\nu\},$$

$$K'_0, K'_1 \xleftarrow{\text{U}} \mathbb{G}_T,$$

$$C_{0,j} \xleftarrow{\text{U}} \mathbb{G}_T \text{ for } j \in [1, \nu-1], \quad C_{0,j} \leftarrow K'_\delta \cdot K_j \text{ for } j = [\nu+1, \widehat{\ell_1}],$$

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

$$\widehat{\overrightarrow{\mathsf{Hdr}}} = \left((\mathsf{Hdr}_j, C_{0,j})_{j=1}^{\widehat{\ell_1}}, (\mathsf{tag}_i)_{i=1}^{\widehat{m_2}}\right).$$

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

[AMORH14] Gora Adj, Alfred Menezes, Thomaz Oliveira, and Francisco Rodrguez-Henrquez. 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/`.

[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.

[BGJT13] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. *CoRR*, abs/1306.4244, 2013.

[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.

[CM11] Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings – the role of ψ 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.

[Del07] Cécile Delerablée. Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys. In *ASIACRYPT*, pages 200–215, 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.

[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 $gf(2^9234)$. `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.

[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.

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

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. http://eprint.iacr.org/.

[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.

[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.