Reinventing BEDs

Formal Treatment of Broadcast Encryption with Dealership
and

Practical Constructions

Sayantan Mukherjee[1] and Avishek Majumder[2]

[1] University of St Gallen, Switzerland ⋆
[2] ISI Kolkata, India ⋆⋆

Abstract. Broadcast Encryption allows a sender to send a message to more than one receiver. In a typical broadcast encryption, the broadcaster decides the privileged set as in who all can decrypt a particular ciphertext. Gritti et al. (IJIS'16) introduced a new primitive called Broadcast Encryption with Dealership (BED), where the dealer/wholesaler decides the privileged set. This rather recently introduced primitive allows a wholesaler to buy content from the broadcaster and sell it to users. Following their construction, to date, three more constructions of *broadcast encryption with dealership* have been proposed. Among them, the first showed the BED construction of Gritti et al. (IJIS'16) to be insecure.

All the state-of-the-arts works were unable to fully identify the requirements of a BED scheme. We first identify and propose a new security requirement that has not been considered before. After formally defining a BED scheme, we show simple pairing-based attacks on all previous constructions rendering all of them useless. We then give the first secure BED construction in the composite-order pairing groups. This construction achieves constant-size ciphertext and secret keys but achieves selectively secure message hiding only. We then give our second construction from Li and Gong's (PKC'18) anonymous broadcast encryption. This construction achieves adaptively secure message hiding but has ciphertext size dependent on the size of the privileged set. Following that, we propose our third and final construction that achieves constant size ciphertext in the standard model and achieves adaptive message hiding security.

1 Introduction

Public key encryption (PKE) allows two parties (say Alice and Bob) to communicate securely without any shared secret key. In this setting, the receiver Alice publishes her *public key* which the sender Bob uses to encrypt his message. The security of a PKE ensures none, but Alice decrypts the message. Now consider the scenario where Bob wants to send a message to a set of users. Broadcast Encryption (BE) [FN94,BGW05] addresses this problem in the literature. In broadcast encryption (BE) a message is encrypted for a set S called *privileged set*. Any user from the privileged set (so-called privileged user) can decrypt the ciphertext. A broadcast encryption (BE) scheme is considered to be secure if all the users outside S together can not decrypt the message. A BE scheme is called fully secure if an adversary chooses S adaptively in the security game. BE is selective secure if the adversary has to submit the set S at the beginning before even seeing the parameters. The major setback of BE is that it can not accommodate any number of new users joining the system without changing the public parameters. This problem was alleviated by the introduction of *Identity-Based Broadcast Encryption* (IBBE) [Del07,RS16]. IBBE brings a lot more dynamism in the sense it supports adding new users to the system without changing the system public key.

Along with security, *privacy* has always been at the center of cryptography. In broadcast encryption, the usual practice is to publish the privileged set description along with the ciphertext. This naturally brings about an obvious question if a broadcast encryption *can also hide the privileged set*? This finds application

⋆ csayantan.mukherjee@gmail.com
⋆⋆ avishek.majumder1991@gmail.com

in protocols that requires the privacy of receivers. To answer this question the notion of *Anonymous Broadcast Encryption* (anon-BE) [BBW06,LPQ12,LG18] was introduced. As discussed above, the privileged set description for which the ciphertext has been constructed is hidden in an anon-BE.

Consider a related scenario where an organization like Pay TV wants to broadcast digital content to a large group of people. It's often become challenging for a single organization to maintain such a user base. One solution for this is that the main organization sells its digital content to some sub-distributor (we call them *dealer* in our work). Users register with a dealer and purchase their subscription. The dealer then buys digital content in bulk and broadcasts the encrypted content to the subscribed users. In literature, this variant of BE was introduced by Gritti et al. [GSP$^+$16] and was named the *Broadcast Encryption with Dealership* (BED).

As motivated above, there are three major entities in a typical BED scheme: broadcaster, dealers, and end-users. Users who want to get digital content from a particular broadcaster get themselves registered with the respective dealers. Among the registered users, those who purchase a subscription, we call them *privileged users*. The dealer buys digital content for the privileged user set S. The broadcaster encrypts its digital content for users in S, which the dealer broadcasts. Being a variant of BE, a BED introduces several new concerns about security. Firstly, a BED needs the unprivileged end-users (who are not part of the privileged set) cannot decrypt the ciphertext. Since the BED accommodates more entities than a typical BE scheme, Gritti et al. in [GSP$^+$16] introduced some more security requirements. To name a few, a typical BED scheme needs to be secure according to so-called *group privacy*, *maximum user accountability* etc. In this work, we notice a gap in the security requirements and introduce another security requirement called *message indistinguishability from dealer*. Next, we discuss all the security notions informally.

Recall that, in the BED scheme, a dealer decides a privileged set S, and the broadcaster encrypts the message for the set S. Therefore, it is necessary to keep the privileged set S hidden from the broadcaster; otherwise, the broadcaster can serve the users in S directly (whenever possible), thereby destroying the dealer's business. Thus the dealer, instead of giving the whole set S gives a some-what proof of the set S. The dealer does that by providing a group token $\mathit{\Gamma}_S$ it computed for the privileged set S. We need this group token to achieve the following properties:

(1) It hides the set S.
(2) Cardinality of S can be verified from $\mathit{\Gamma}_S$.
(3) Given $\mathit{\Gamma}_S$, the broadcaster can produce ciphertext CT_S for S without explicitly knowing S.
(4) Dealer cannot infer any information about the underlying message from the ciphertext CT_S the broadcaster generates using $\mathit{\Gamma}_S$.

Other than the above, we also need that (5) in a BED unprivileged users (i.e., $u \notin S$) cannot infer any information about the underlying message from the ciphertext CT_S. The existing literature already has formulated security games capturing some of these requirements. Precisely, in the literature, (1) is called the *group privacy* security, (2) is formulated as *maximum user accountability* security and (5) is called the *message indistinguishability from unprivileged users*. Furthermore, the existing literature implicitly assumes (3) as a functionality requirement. In this work, we introduce (4) as the *message indistinguishability from dealer* security.

To keep the presentation simpler, we summarize the security requirements informally in Table 1. At the same time, we use the table to justify the motivation of this work given the related works. Looking ahead, the following table shows the inadequacy of the existing works, which we will elaborate on while discussing our related work section. For the time being, we state that we could mount concrete attacks on the *group privacy* of all the existing works and could also find inaccuracies in the security argument of *maximum user accountability* in all of them. Furthermore, the *message indistinguishability from dealer* which was not considered any of the existing works, is marked in Table 1 as "Undefined". The table, therefore, indicates that none of the existing works achieve the security guarantees a BED should possess.

We now motivate the requirement of message indistinguishability from the dealer informally. Suppose a BED scheme where the broadcaster gets a group token $\mathit{\Gamma}_S$ from the dealer for the set S. Broadcaster multiplies the message to one of the components of $\mathit{\Gamma}_S$ and returns it to the dealer. Precisely, let $\mathit{\Gamma}_S = (\mathsf{Hdr}_S, \kappa_S)$ due to

Security Model	Informal Description	Existing Works [AD16,AD17,KLEL18]
Group Privacy	Adversary cannot distinguish between two privileged sets of equal size.	Insecure.
Maximum User Accountability	Adversary cannot pass verification for a privileged set larger than committed size.	Incorrect Proof.
Message Indistinguishability from Dealer	Dealer cannot distinguish between two ciphertext generated for the same privileged set.	Undefined.
Message Indistinguishability from Unprivileged Users	Any unprivileged user cannot distinguish between two ciphertext generated for the same privileged set.	Secure.

Table 1. Broadcast Encryption with Dealership State-of-the-Art.

the underlying broadcast encryption. The broadcaster returns $\mathsf{CT}_S = (\mathsf{Hdr}_S, M \cdot \kappa_S)$. Note that the view of dealer (who provides the group token Γ_S and gets CT_S) and the view of an unprivileged user is quite different. Going by the security models considered in the existing schemes [AD16,AD17,KLEL18], this scheme should be regarded as secure. However, the dealer can simply cancel κ_S to get the message M and sell it to any number of users it likes. This is why critical consideration of *message indistinguishability from dealer* is necessary. Thus we introduce the message indistinguishability from the dealer security model to bridge this gap.

1.1 Related Works

We give a brief overview of the claimed achievements of the existing works next. Being a newly introduced primitive, the literature for BED is not very vast. There have been a few works available on this primitive, [GSP$^+$16,AD16,AD17,KLEL18] to name a few. Gritti et al. [GSP$^+$16] was first to introduce the notion of BED. The idea of BED was very much influenced by the idea of *"membership encryption"* of [GMSV13]. The authors of [GSP$^+$16] argued *group privacy* security of their construction to be *"unconditionally secure"*. To which Acharya and Dutta found an attack and provided the first "secure" construction of BED in [AD16]. The construction of [AD16] and following constructions [AD17,KLEL18] claimed to have achieved group privacy security under the standard discrete-log assumption. Coming to *maximum user accountability* security, construction of [GSP$^+$16] depends on an honest user to report a dishonest dealer. The work of [AD16] overcame this limitation injecting a computationally hard problem i.e. if a dealer can break maximum user accountability, we can solve a computationally hard problem. As a result, [AD16] and the following works [AD17,KLEL18], the broadcaster can detect a dishonest dealer without any assistance from a user. All three works [AD16,AD17,KLEL18] achieved maximum user accountability under parameterized DHE assumption of [GSP$^+$16], and [Cam13]. *Message indistinguishability from unprivileged users* security of [GSP$^+$16] achieved was in *"semi-static"* model, introduced by Gentry and Waters [GW09]. Gentry and Waters in [GW09] provide a generic technique to convert a semi-static scheme to an adaptive secure scheme. Gritti et al. mentioned that the conversion of [GW09] applies for their construction also. The construction of [AD16] was heavily influenced by the broadcast encryption of [Del07]. The *Message indistinguishability from unprivileged users* security of [AD16] was achieved in selective security model under the parameterized GDDHE assumption [Del07]. To improve upon the message indistinguishability from unprivileged users security the same group of authors proposes an adaptive secure BED in [AD17] based on [RWZ12] under the parameterized DABDHE assumption. Kim et al. [KLEL18] modified the construction of [AD17] to support recipient revocation without compromising upon security. All the above constructions achieved their respective property keeping the ciphertext size constant.

1.2 Our Contributions

Our contributions in this paper are many-fold. Firstly, none of the previous works on BED (to the best of our knowledge) considered the dealer's presence with sufficient formalization. This paper first puts forward the security definition of Broadcast Encryption with Dealership addressing the presence of a dealer with sufficient formalization. Our definition, therefore, generalizes all the earlier works [GSP$^+$16,AD16,AD17,KLEL18]. The introduction of a new security game is justified as we observed the view of a dealer and that of an unprivileged user are different. Hence, we consider message indistinguishability from the dealer separately. This bridges a gap that was not considered earlier.

We then present concrete attacks on the *group privacy* of the existing schemes [AD16][AD17][KLEL18]. As we mentioned above, all the above works claimed that their constructions achieve the *group privacy* security under the standard discrete-log problem. However, we mount simple pairing-based attacks rendering all those schemes insecure in the said security model. Moreover, we also identify the errors in the *maximum users accountability* security proof of all the existing schemes.

Then, we propose three constructions that achieve all the desired security guarantees. Our first construction of BED is based on the broadcast encryption of Ramanna et al. proposed in [GLR18]. This construction achieves constant-size ciphertext, which essentially follows from the efficient BE construction of [GLR18]. The main point of interest being BE of [GLR18] does not have anonymous security. Our novelty in our first construction is primarily that we still manage to get *group privacy* (priv) security under the standard assumption DDH.

The above construction, however, is constructed in composite-order pairings and achieves only selective *message indistinguishability from unprivileged users*. We, therefore, propose a new BED from the anonymous broadcast encryption of Li and Gong [LG18] along with a key-binding [Fis99] symmetric encryption. Our intuition here was, an anonymous BE hides the privileged set. So we can get the group privacy security and message indistinguishability from unprivileged users for free while using an anon-BE. On top of that, anon-BE of [LG18] is *cardinality-revealing* i.e., the ciphertext size gives away the cardinality of S. We show that this BED construction also achieves message indistinguishability from dealers.

Here we note that the above construction outputs large ciphertext, namely, the ciphertext size depends on the privileged set size. To improve upon this, we propose a new BED from the broadcast encryption of Ren et al. [RWZ12]. This construction achieves constant size ciphertext and still achieves adaptive message indistinguishability from unprivileged users. Again, we note that BE of [RWZ12] does not have anonymous security, but we still manage to get *group privacy* (priv) security under the standard assumption DDH.

To summarize, below, we provide a concise list of our contributions toward making the first secure BED scheme.

1. Firstly, we have formalized the existing definition by bridging a gap in the security requirements.
2. Then we have shown simple pairing-based attacks on all the existing schemes [AD16,AD17,KLEL18] rendering them insecure in the *group privacy* game.
3. We further have shown that the security proof for *maximum user accountability* is incorrect.
4. This is when we propose our first secure construction that is achieved from [GLR18]. This construction uses composite order bilinear pairing to achieves only selective message indistinguishability from unprivileged users.
5. To improve upon the security, we propose our second construction from the anonymous broadcast encryption of [LG18].
6. Our second construction results in a larger ciphertext size. We, therefore, propose our final construction that achieves constant-size ciphertext and adaptive message indistinguishability from unprivileged users that too in the prime-order bilinear pairing groups.

1.3 Our Techniques

In this section, we informally discuss our techniques. Firstly, as we pointed out above, we could mount simple pairing-based attacks on the *group privacy* all the existing schemes. Informally speaking, these attacks verifies if the public key and the group token result in a DDH instance. To alleviate this, we take inspiration from [Duc10] and resort to the asymmetric bilinear groups. More precisely, we introduce extra randomness to prove the *group privacy* security under the standard DDH assumption in the asymmetric bilinear group.

For the *maximum user accountability* security, however, we could not find any measure to give a standard assumption-based security proof. We give a *maximum user accountability* security proof in the generic group model. Intuitively, the challenger gets to know all the group-based computations an adversary evaluates in the generic group model. Therefore, we can extract certain knowledge that we could not manage in the standard model.

Notice that the *message indistinguishability from dealer* security, although looks very similar to blindness security of blind signature [FHS15], there is a small difference which turns out to be quite complex for security reduction to go through. In *message indistinguishability from dealer*, the adversary computes a group token completely on its own and hands it over to the challenger along with two messages of its choice; the challenger, in turn, returns encryption of one message. As the adversary chooses the privileged set and the randomness used in the group token generation, simulation even using interactive assumptions turns out to be quite complicated. We again choose the generic group model to justify our constructions achieve *message indistinguishability from dealer*.

Along with these, we also have modified the security proof of [GLR18] to work in composite-order asymmetric bilinear groups for two primes. Earlier, the security proof of [GLR18] needed symmetric pairing composite-order groups of three primes.

1.4 Organization of Our Paper

In Section 2, we present the definitions and mathematical preliminaries. In Section 3, we present a formal definition of Broadcast Encryption with Dealership (BED). In Section 4, we then show that all the available BED schemes are insecure. This is followed by Section 5 where we give the first concrete construction of BED with constant-size ciphertext but in a composite-order pairing setting. Then in Section 6, we give a prime-order BED construction achieving stronger security. We give our final construction of BED in Section 7 which also is instantiated in the prime-order pairing groups, achieves stronger security, and still has constant-size ciphertext. Then, we conclude this paper in Section 8. As, in this paper, we introduced an asymmetric version of weaker augmented bilinear Diffie-Hellman exponent assumption (wABDHE) from [RWZ12], we argue it's security in Appendix A.

2 Definitions and Preliminaries

We start by defining some necessary tools that would be required for our construction.

2.1 Notation

For $a, b \in \mathbb{N}$ such that $a \leq b$, we often use $[a, b]$ to denote $\{a, \ldots, b\}$. For a set X, we write $x \xleftarrow{\$} X$ to say that x is a uniformly random element of X. The ppt abbreviation stands for probabilistic polynomial time. For any algorithm $\mathcal{A}$, oracle $\mathcal{O}$ and problem instance $\mathsf{problem}$, $\mathcal{A}^{\mathsf{problem}} \Rightarrow 1$ denotes that $\mathcal{A}$ given the problem instance $\mathsf{problem}$ outputs 1 and $\mathcal{A}^{\mathcal{O}} \Rightarrow 1$ denotes that $\mathcal{A}$ given the oracle access $\mathcal{O}$ outputs 1. Let $\mathsf{neg} : \mathbb{Z}_p \to \mathbb{R}^+$ is called a negligible function if for all positive polynomial $p(\cdot)$ and for sufficiently large values of x, $\mathsf{neg}(x) < 1/p(x)$.

2.2 Groups and Hardness Assumptions

This section presents a discussion about different types of elliptic curve groups and hardness assumptions that we will require in this work.

2.2.1 Prime Order Groups and Hardness Assumptions Let $\mathcal{G} = (p, g, \mathbb{G}) \leftarrow \mathsf{PGen}$ be the output of prime order group generator where $\mathbb{G} = \langle g \rangle$ is a cyclic group of order p where p is a large prime number.

DDH Assumption (DDH)**.** The decisional Diffie-Hellman problem (DDH) in the group $\mathbb{G}$ is defined as follows.

Definition 1. *Given $\mathcal{G} = (p, g, \mathbb{G}) \leftarrow \mathsf{PGen}(1^\lambda)$ and $X = (\mathcal{G}, g, g^a, g^b, g^c)$ we say that the Decisional Diffie-Hellman assumption (DDH) holds in $\mathcal{G}$ if for all* ppt *adversary $\mathcal{A}$ the advantage $\mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{A},\mathcal{G}}(\lambda)$ defined below is* $\mathsf{neg}(\lambda)$.

$$\left|\Pr\left[\mathcal{A}(g, h, g^a, g^b, g^{ab}) = 1\right] - \Pr\left[\mathcal{A}(g, h, g^a, g^b, g^c) = 1\right]\right|$$

where the probability is taken over $\mathcal{G} \overset{\$}{\leftarrow} \mathsf{PGen}(1^\lambda)$, $a, b, c \overset{\$}{\leftarrow} \mathbb{Z}_p$ and the random coin consumed by $\mathcal{A}$.

We next consider a set of elliptic curve groups where bilinear pairing function is efficiently evaluated.

Bilinear Pairing. Let $\mathbb{G}$, $\mathbb{H}$, $\mathbb{G}_T$ be three commutative multiplicative groups. A map $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$ is called an admissible bilinear pairing if,

- (Bilinear) For all $g \in \mathbb{G}$ and all $h \in \mathbb{H}$, $e(g^a, h^b) = e(g, h)^{ab}$ for any $a, b \in \mathbb{N}$.
- (Non-degenerate) $e(g, h) = 1$ only if $g = 1$ or $h = 1$.
- (Computable) For all $g \in \mathbb{G}$ and all $h \in \mathbb{H}$, there is a ppt algorithm that computes $e(g, h)$.

Bilinear pairings are of three kinds. A bilinear pairing is called a Type-1 pairing when $\mathbb{G} = \mathbb{H}$. Now if we have $\mathbb{G} \neq \mathbb{H}$ but there is a known isomorphism between $\mathbb{G}$ and $\mathbb{H}$ it is regarded as Type-2 pairing. In this work, we used so-called Type-3 pairing where $\mathbb{G}$ and $\mathbb{H}$ have no known isomorphism.

2.2.2 Prime Order Asymmetric Bilinear Pairing The prime order asymmetric bilinear group generator PBGen, takes security parameter 1^λ as input and outputs a septenary tuple $(p, g, h, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e)$ where all of $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{G}_T$ are cyclic groups of order large prime p, $\mathbb{G} = \langle g \rangle$, $\mathbb{H} = \langle h \rangle$ and $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$ is an admissible, non-degenerate asymmetric bilinear pairing. In this work, we make use of the following hardness assumptions.

Decisional Diffie-Hellman Assumption (DDH)**.**

Definition 2. *Given $\mathcal{PG} = (p, g, h, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) \leftarrow \mathsf{PBGen}(1^\lambda)$ and $X = (\mathcal{PG}, g, h, g^a, g^b, g^c)$ we say that the Decisional Diffie-Hellman assumption (DDH) holds in $\mathcal{PG}$ if for all* ppt *adversary $\mathcal{A}$ the advantage $\mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{A},\mathcal{PG}}(\lambda)$ defined below is* $\mathsf{neg}(\lambda)$.

$$\left|\Pr\left[\mathcal{A}(g, h, g^a, g^b, g^{ab}) = 1\right] - \Pr\left[\mathcal{A}(g, h, g^a, g^b, g^c) = 1\right]\right|$$

where the probability is taken over $\mathcal{PG} \overset{\$}{\leftarrow} \mathsf{PBGen}(1^\lambda)$, $a, b, c \overset{\$}{\leftarrow} \mathbb{Z}_p$ and the random coin consumed by $\mathcal{A}$.

Weaker asymmetric augmented bilinear Diffie-Hellman exponent assumption (waABDHE)**.** Here we introduce an asymmetric version of weaker augmented bilinear Diffie-Hellman exponent assumption (wABDHE) introduced in [RWZ12].

Definition 3. *We say that the* waABDHE *assumption holds relative to $\mathcal{PG} = (p, g, h, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) \leftarrow \mathsf{PBGen}(1^\lambda)$ given $\kappa_{g,h,g',h',\alpha,q} = (g', h', g^\alpha, \ldots, g^{\alpha^q}, h^\alpha, \ldots, h^{\alpha^q}, g'^{\alpha^{q+2}}, \ldots, g'^{\alpha^{2q}}, h'^{\alpha^{q+2}}, \ldots, h'^{\alpha^{2q}}, Z) \in \mathbb{G}^{2q+1} \times \mathbb{H}^{2q+1} \times \mathbb{G}_T$, if for all* ppt *adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{waABDHE}}_{\mathcal{A},\mathcal{PG}}(\lambda)$ defined below is* $\mathsf{neg}(\lambda)$,

$$\left|\Pr\left[\mathcal{A}(g, h, \kappa_{g,h,g',h',\alpha,q}, Z) = 1\right] - \Pr\left[\mathcal{A}(g, h, \kappa_{g,h,g',h',\alpha,q}, T) = 1\right]\right|$$

where the probabilities are taken over $\mathcal{PG} \overset{\$}{\leftarrow} \mathsf{PBGen}(1^\lambda)$, $\alpha, \lambda \overset{\$}{\leftarrow} \mathbb{Z}_p$, $g' = g^\lambda, h' = h^\lambda$, $Z = e(g', h)^{\alpha^{q+1}}$ or $e(g, h')^{\alpha^{q+1}}$ and $T \in \mathbb{G}_T$ and the random coin consumed by $\mathcal{A}$.

2.2.3 Composite Order Asymmetric Bilinear Pairing A composite order asymmetric bilinear group generator CBGen, takes the security parameter 1^λ as input and returns a 10-tuple $\mathcal{CG} = (p, q, g_p, g_q, h_p, h_q, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e)$ where both $\mathbb{G}, \mathbb{H}$ are cyclic groups of order $N = pq$ where both p and q are large primes, g_p, h_p are elements of order p, g_q, h_q are elements of order q, and $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$ is an admissible, non-degenerate bilinear map. Note that $\mathbb{G}$ has two subgroups $\mathbb{G}_p$ and $\mathbb{G}_q$. Similarly, $\mathbb{H}$ has two subgroups $\mathbb{H}_p$ and $\mathbb{H}_q$.

DDH Assumption in Subgroup $\mathbb{G}_p$.

Definition 4. *Given $\mathcal{CG} = (p, q, g_p, g_q, h_p, h_q, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) \leftarrow \mathsf{CBGen}(1^\lambda)$ and $X = (g_p, g_q, h_p, h_q, g_p^a, g_p^{ab}, g_p^c)$ we say that the Decisional Diffie-Hellman assumption holds in $\mathcal{CG}$ if for all* ppt *adversary $\mathcal{A}$ the advantage $\mathsf{Adv}^{\mathsf{DDH}_{\mathcal{C}}}_{\mathcal{A},\mathcal{CG}}(\lambda)$ defined below is* $\mathsf{neg}(\lambda)$.

$$\left|\Pr\left[\mathcal{A}(g_p, g_q, h_p, h_q, g_p^a, g_p^{ab}, g_p^b) = 1\right] - \Pr\left[\mathcal{A}(g_p, g_q, h_p, h_q, g_p^a, g_p^{ab}, g_p^c) = 1\right]\right|$$

where the probability is taken over $\mathcal{CG} \xleftarrow{\$} \mathsf{CBGen}(1^\lambda)$, $a, b, c \xleftarrow{\$} \mathbb{Z}_p$ and the random coin consumed by $\mathcal{A}$.

Subgroup Decision Assumptions.

Definition 5. *Given $\mathcal{CG} = (p, q, g_p, g_q, h_p, h_q, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) \leftarrow \mathsf{CBGen}(1^\lambda)$ and $W = (g_p, h_p, Z)$ we say that the AS_1 assumption holds in $\mathcal{CG}$ if for all* ppt *adversary $\mathcal{A}$ the advantage $\mathsf{Adv}^{\mathsf{AS}_1}_{\mathcal{A},\mathcal{CG}}(\lambda)$ defined below is* $\mathsf{neg}(\lambda)$.

$$\left|\Pr\left[\mathcal{A}(g_p, h_p, g_p^a) = 1\right] - \Pr\left[\mathcal{A}(g_p, h_p, g_p^a g_q^b) = 1\right]\right|$$

where the probability is taken over $\mathcal{CG} \xleftarrow{\$} \mathsf{CBGen}(1^\lambda)$, $g_p^a \xleftarrow{\$} \mathbb{G}_p$, $g_p^a g_q^b \xleftarrow{\$} \mathbb{G}$ and the random coin consumed by $\mathcal{A}$.

Definition 6. *Given $\mathcal{CG} = (p, q, g_p, g_q, h_p, h_q, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) \leftarrow \mathsf{CBGen}(1^\lambda)$ and $W = (g_p, h_p, X, Y, Z)$ we say that the AS_2 assumption holds in $\mathcal{CG}$ if for all* ppt *adversary $\mathcal{A}$ the advantage $\mathsf{Adv}^{\mathsf{AS}_2}_{\mathcal{A},\mathcal{CG}}(\lambda)$ defined below is* $\mathsf{neg}(\lambda)$.

$$\left|\Pr\left[\mathcal{A}(g_p, h_p, h_q, g_p^a g_q^b, h_p^c) = 1\right] - \Pr\left[\mathcal{A}(g_p, h_p, h_q, g_p^a g_q^b, h_p^c h_q^d) = 1\right]\right|$$

where the probability is taken over $\mathcal{CG} \xleftarrow{\$} \mathsf{CBGen}(1^\lambda)$, $g_p^a g_q^b \xleftarrow{\$} \mathbb{G}$, $h_p^c \xleftarrow{\$} \mathbb{H}_p$, $h_p^c h_q^d \xleftarrow{\$} \mathbb{H}$ and the random coin consumed by $\mathcal{A}$.

2.2.4 Generic Group Model Generic group model was explored formally first by Shoup [Sho97]. This technique is used to prove the lower bounds of certain computational and decisional problems. This is explored in terms of the computational power of any *generic algorithm* against the targeted problems. A generic algorithm only assumes that each group element is uniquely encoded and does not exploit any other properties of the underlying group structure.

2.3 Symmetric Key Encryption

A symmetric encryption scheme SE with keyspace $\mathcal{K}$ consists of two algorithms (E, D)

- $\mathsf{E}(\kappa, M)$: It takes the secret key $\kappa \in \mathcal{K}$ and a message M. Outputs the ciphertext ct.
- $\mathsf{D}(\kappa, \mathsf{ct})$: It takes the secret key $\kappa \in \mathcal{K}$ and a ciphertext ct. Outputs the message M or $\perp$ if the decryption fails.

The correctness can be stated as follows: for all $\kappa \in \mathcal{K}$ and all message M, we have $\mathsf{D}(\kappa, \mathsf{E}(\kappa, M)) = M$ with overwhelming probability.

We reproduce the security definition from [LG18]. A symmetric encryption scheme SE satisfies indistinguishability under chosen plaintext attack ($\mathsf{ind\text{-}cpa}$) if for all ppt adversaries $\mathcal{A}$,

$$\mathrm{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathcal{A},\mathsf{SE}}(\lambda) = \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{SE}}(1^\lambda, \mathcal{A}) = 1\right] \leq \mathsf{neg}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{SE}}(1^\lambda, \mathcal{A})$ is defined in Figure 1.

In this work, we require the symmetric encryption SE to be *key-binding* [Fis99]. For any message M and any two distinct secret keys $\kappa, \kappa' \in \mathcal{K}$, $\mathsf{SE.D}(\kappa', \mathsf{SE.E}(\kappa, M)) = \perp$. See [Fis99] for details.

Game Description
$\underline{\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{SE}}(1^\lambda, \mathcal{A})}$
$\mathcal{G} \leftarrow \mathsf{GGen}(1^\lambda)$
$(M_0, M_1) \leftarrow \mathcal{A}(1^\lambda, \mathcal{K})$
$\beta \xleftarrow{\$} \{0,1\}$, $\kappa \xleftarrow{\$} \mathcal{K}$
$\mathsf{ct}_\beta \leftarrow \mathsf{E}(\kappa, M_\beta)$
Return 1 if $\beta = \beta'$ where $\beta' \leftarrow \mathcal{A}(\mathsf{ct}_\beta)$

Fig. 1. IND-CPA for SE

3 Broadcast Encryption with Dealership

The introductory discussion of this paper shows that in none of the previous works [GSP$^+$16,AD16,AD17] [KLEL18] the notion of BED was considered with much detail. It is a necessity to formalize any primitive before diving into construction. This section aims to scrutinize the requirements of a BED scheme, its system requirements, and corresponding security requirements.

We start by informally describing what should be the requirements of a BED scheme. We follow it up with, discussion on the entities of a BED algorithm and the interaction between them. Finally, we formalize the security requirements of a BED scheme. As we mentioned earlier, all the previous works have considered only three types of security for BED, namely *"Group Privacy", "Maximum User Accountability"* and *"Message Indistinguishability from Unprivileged Users"*. We also have justified in the Introduction that hiding the message from the dealer is also important for a BED. Thus a separate security guarantee of message indistinguishability from the dealer is also necessary. Notice that the dealer does not have an initial state (like users have a user secret key). Moreover, the dealer is never a part of the privileged set. However, the dealer has a significant role in the system. The dealer is the one who selects the privileged set S. The dealer also computes the group token (Γ_S) for S. Later this Γ_S is used by the broadcaster in ciphertext generation. More precisely, the broadcaster re-randomizes elements of Γ_S before encrypting the message. The dealer, therefore, gets a ciphertext which is a re-randomized version of what it has generated. Therefore, it is crucial to argue that the dealer can not infer any information from the ciphertext. This discussion shows that the view of a dealer is quite different from an unprivileged user and needs to be considered separately.

As a possible way to deal with this newly introduced entity, i.e., the dealer, we have introduced a new notion of security in this work, that is *"Message Indistinguishability from Dealer"*. This security notion guarantees that a dealer can not distinguish between two ciphertexts generated from the same group token. This new notion of security guarantees that the dealer can not re-broadcast the content.

3.1 System Model

Entities of a BED system are key generation center (KGC), broadcaster(s), dealer(s), and end-users. KGC generates a public key (pk), master secret key (msk), and user secret key (sk_i). KGC publishes the public key, keeps the master secret key to itself, and distributes the user secret keys when invoked with a join request. The broadcaster is the one with the digital content (message) that it wants to broadcast. Usually, a broadcaster is a large organization that wants to sell digital content, and dealers are some sub-distributors who in turn sell the digital content for some incentive. A dealer buys ciphertext from the broadcaster for a privileged set of its choice. Broadcaster produces the ciphertext.

In the following section, we mention the roles of every entity of a BED model. Our considered model for this work assumes no collusion between *broadcaster-users* or *dealer-user*. Whereas users (more specifically, unprivileged users) can collude between themselves. Whenever a dealer buys ciphertext, it also commits to a value k for the maximum size of the set S. The dealer in our model of BED is *"semi-honest but curious"*[3]. Below, we give a complete run of the protocol in terms of the interaction between the entities.

[3] A semi-honest but curious adversary computes the group token honestly by following the protocol. An adversary can try to be dishonest about the committed set size k.

1. At the beginning, a key generation center runs $\mathsf{BED.Setup}$ taking as input the security parameter λ and the maximal size of the set of users n and produces $(\mathsf{pk}, \mathsf{msk})$. It publishes the public parameters pk and keeps the master secret key msk to itself.
2. When invoked with a join request for a user i, KGC computes the user's secret key sk_i using $\mathsf{BED.KeyGen}$ for user $i \in [n]$ and send the user's secret key via a secure channel.
3. The dealer selects a group of the user $S \subseteq [n]$ of size k'.
4. The dealer generates a group token $\mathit{\Gamma}_S$ using $\mathsf{BED.GroupGen}$ and gives it to the broadcaster along with the value k such that $k' = |S| \leq k \leq n$. The dealer sends S to every user in the privileged set via a secure channel.
5. The broadcaster verifies the group token with $\mathsf{BED.Verify}$ and upon verification generates ciphertext CT_S using $\mathit{\Gamma}_S$ for the set S and broadcast CT_S.
6. Any end-user can decrypt the message M using their secret key sk_i if $i \in S$.

3.2 Definition

A broadcast encryption with dealership framework is a three-party[4] protocol where a dealer acts as a middleman in between broadcaster and end-users. Precisely, a dealer buys digital content for a privileged set to serve the end-users. We give a formal definition of BED next refurbished from the earlier works [GSP+16,AD16,AD17,KLEL18]. A BED is a tuple of six ppt algorithms $\mathsf{BED} = (\mathsf{BED.Setup}, \mathsf{BED.KeyGen}, \mathsf{BED.GroupGen}, \mathsf{BED.Verify}, \mathsf{BED.Encrypt}, \mathsf{BED.Decrypt})$

- $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{BED.Setup}(1^\lambda, n)$: It takes input the maximal size of the set of receivers n and the security parameter λ and outputs public parameter pk and a master secret key msk.
- $(\mathsf{sk}_i) \leftarrow \mathsf{BED.KeyGen}(\mathsf{msk}, i)$: On invocation with $\mathsf{pk}, \mathsf{msk}$ and a user identity i, it outputs secret key sk_i for user i.
- $(\mathit{\Gamma}_S, k) \leftarrow \mathsf{BED.GroupGen}(\mathsf{pk}, k, S)$: It takes input a set of users $S \subseteq [n]$ of size k' and a threshold value k such that $|S| = k' \leq k$ where k is the (maximum) number of users the dealer wishes to serve. It returns a tuple $(\mathit{\Gamma}_S, k)$ where $\mathit{\Gamma}_S$ is a group token for the set S.
- $(0 \vee 1) \leftarrow \mathsf{BED.Verify}(\mathsf{pk}, \mathit{\Gamma}_S, k)$: On input pk, a group token $\mathit{\Gamma}_S$ and a number $k \in [n]$ which is the (maximum) number of users for which this token is created, it verifies whether $|S| \leq k$ or not.

$$\mathsf{BED.Verify}(\mathit{\Gamma}_S, \mathsf{pk}, k) = \begin{cases} 1, & \text{if } |S| \leq k \\ 0, & \text{otherwise.} \end{cases}$$

- $(\mathsf{CT}_S) \leftarrow \mathsf{BED.Encrypt}(\mathsf{pk}, \mathit{\Gamma}_S, M)$: It takes the public key pk and a verified group token $\mathit{\Gamma}_S$ and a message M and outputs ciphertext CT_S.
- $(M) \leftarrow \mathsf{BED.Decrypt}(\mathsf{pk}, \mathsf{sk}_i, (S, \mathsf{CT}_S))$: On input $\mathsf{pk}, \mathsf{sk}_i$ and a cipher text CT_S for a set S, $\mathsf{BED.Decrypt}$ outputs message M if $i \in S$.

Correctness. A BED scheme is said to be correct if $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{BED.Setup}(1^\lambda, n)$, for all $S \subset [n]$, $(\mathit{\Gamma}_S, k) \leftarrow \mathsf{BED.GroupGen}(\mathsf{pk}, k, S)$, and $\mathsf{CT}_S \leftarrow \mathsf{BED.Encrypt}(\mathsf{pk}, \mathit{\Gamma}_S, M)$ then for all $i \in S$, the following condition holds:

$$\mathsf{BED.Decrypt}(\mathsf{pk}, \mathsf{BED.KeyGen}(\mathsf{msk}, i), (S, \mathsf{CT}_S)) = M.$$

3.3 Security Definition

So far, we have discussed the security requirements of BED informally. The following section formalizes those security notions. To an extent, we would follow the definition given by [AD16]. We would modify or add to their definition as required.

[4] Precisely, BED accommodates three different entities.

3.3.1 Group Privacy

This security model captures the requirement that from a group token $\mathit{\Gamma}_S$, the broadcaster does not get any information but the cardinality of the underlying set $S \subseteq [n]$. A BED scheme BED satisfies group privacy (priv) if for all ppt adversaries $\mathcal{A}$,

$$\mathrm{Adv}^{\mathsf{priv}}_{\mathcal{A},\mathsf{BED}}(\lambda) = \Pr\left[\mathsf{Exp}^{\mathsf{priv}}_{\mathsf{BED}}(1^\lambda, \mathcal{A}) = 1\right] \leq \mathsf{neg}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{priv}}_{\mathsf{BED}}(1^\lambda, \mathcal{A})$ is defined in Figure 2.

Game Description
$\underline{\mathsf{Exp}^{\mathsf{priv}}_{\mathsf{BED}}(1^\lambda, \mathcal{A})}$
$\mathcal{G} \leftarrow \mathsf{GGen}(1^\lambda)$
$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{BED.Setup}(1^\lambda, n)$
$(S_0, S_1) \leftarrow \mathcal{A}(\mathsf{pk})$ s.t. $\lvert S_0\rvert = \lvert S_1\rvert = k \leq n$
Sample $\beta \xleftarrow{\$} \{0,1\}$, $\mathit{\Gamma}_{S_\beta} \leftarrow \mathsf{BED.GroupGen}(\mathsf{pk}, k, S_\beta)$
$\beta' \leftarrow \mathcal{A}(\mathsf{pk}, \mathit{\Gamma}_{S_\beta})$
Return 1 if $\beta = \beta'$.

Fig. 2. Group Privacy for BED

3.3.2 Maximum User of Accountability

This security model captures the requirement that a group token $\mathit{\Gamma}_S$ can not encode a privileged set S of size bigger than its claimed size k. A BED scheme BED satisfies maximum user of accountability (mua) if for all ppt adversaries $\mathcal{A}$,

$$\mathrm{Adv}^{\mathsf{mua}}_{\mathcal{A},\mathsf{BED}}(\lambda) = \Pr\left[\mathsf{Exp}^{\mathsf{mua}}_{\mathsf{BED}}(1^\lambda, \mathcal{A}) = 1\right] \leq \mathsf{neg}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{mua}}_{\mathsf{BED}}(1^\lambda, \mathcal{A})$ is defined in Figure 3.

Game Description
$\underline{\mathsf{Exp}^{\mathsf{mua}}_{\mathsf{BED}}(1^\lambda, \mathcal{A})}$
$\mathcal{G} \leftarrow \mathsf{GGen}(1^\lambda)$
$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{BED.Setup}(1^\lambda, n)$
$(\mathit{\Gamma}_{S^*}, k) \leftarrow \mathcal{A}(\mathsf{pk})$
Return 1 if the following holds:
$\mathsf{BED.Verify}(\mathsf{pk}, \mathit{\Gamma}_{S^*}, k) \rightarrow 1$ AND $\lvert S^*\rvert > k$

Fig. 3. Maximum User Accountability for BED

3.3.3 Message Indistinguishability for Dealer under CPA

This security model captures the requirement that given a ciphertext CT_S, the dealer can not get any information about the underlying message M. A BED scheme BED satisfies message indistinguishability for dealer ($\mathsf{cpa}_{\mathcal{D}}$) if for all ppt adversaries $\mathcal{A}$,

$$\mathrm{Adv}^{\mathsf{cpa}_{\mathcal{D}}}_{\mathcal{A},\mathsf{BED}}(\lambda) = \Pr\left[\mathsf{Exp}^{\mathsf{cpa}_{\mathcal{D}}}_{\mathsf{BED}}(1^\lambda, \mathcal{A}) = 1\right] \leq \mathsf{neg}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{cpa}_{\mathcal{D}}}_{\mathsf{BED}}(1^\lambda, \mathcal{A})$ is defined in Figure 4.

3.3.4 Message Indistinguishability for Unprivileged Users under CPA

This security model captures the requirement that given a ciphertext CT_S, no unprivileged user can get any information about the underlying message M even if they collude together. A BED scheme BED satisfies message indistinguishability for unprivileged user ($\mathsf{cpa}_{\mathcal{U}}$) if for all ppt adversaries $\mathcal{A}$,

$$\mathrm{Adv}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathcal{A},\mathsf{BED}}(\lambda) = \Pr\left[\mathsf{Exp}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{BED}}(1^\lambda, \mathcal{A}) = 1\right] \leq \mathsf{neg}(\lambda),$$

where $\mathsf{Exp}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{BED}}(1^\lambda, \mathcal{A})$ is defined in Figure 5.

Game Description
$\underline{\mathsf{Exp}_{\mathsf{BED}}^{\mathsf{cpa}_{\mathcal{D}}}(1^\lambda, \mathcal{A})}$
$\mathcal{G} \leftarrow \mathsf{GGen}(1^\lambda)$
$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{BED.Setup}(1^\lambda, n)$
$(\mathit{\Gamma}_S, M_0, M_1) \leftarrow \mathcal{A}(\mathsf{pk})$ for $\lvert S\rvert \leq k \leq n$
where $\mathit{\Gamma}_S \leftarrow \mathsf{BED.GroupGen}(\mathsf{pk}, k, S)$
Sample $\beta \xleftarrow{\$} \{0,1\}$, $\mathsf{CT}_{S,\beta} \leftarrow \mathsf{BED.Enc}(\mathsf{pk}, \mathit{\Gamma}_S, M_\beta)$
$\beta' \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{CT}_{S,\beta})$
Return 1 if $\beta = \beta'$.

Fig. 4. Message Indistinguishability from Dealer for BED

Game Description	Oracle Description
$\underline{\mathsf{Exp}_{\mathsf{BED}}^{\mathsf{cpa}_{\mathcal{U}}}(1^\lambda, \mathcal{A})}$	$\underline{\mathcal{O}_{\mathsf{sk}}(i)}$
$\mathcal{G} \leftarrow \mathsf{GGen}(1^\lambda)$	$\mathcal{Q}_{\mathsf{sk}} \leftarrow \mathcal{Q}_{\mathsf{sk}} \cup \{i\}$
$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{BED.Setup}(1^\lambda, n)$	Run $\mathsf{sk}_i \leftarrow \mathsf{BED.KeyGen}(\mathsf{msk}, i)$
$\mathcal{Q}_{\mathsf{sk}} \leftarrow \phi$	Return sk_i
$(S, M_0, M_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sk}}(\cdot)}(\mathsf{pk})$ such that $\lvert S\rvert \leq k \leq n$	
$\mathit{\Gamma}_S \leftarrow \mathsf{BED.GroupGen}(\mathsf{pk}, k, S)$	
Sample $\beta \xleftarrow{\$} \{0,1\}$, $\mathsf{CT}_{S,\beta} \leftarrow \mathsf{BED.Enc}(\mathsf{pk}, \mathit{\Gamma}_S, M_\beta)$	
$\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sk}}(\cdot)}(\mathsf{pk}, \mathsf{CT}_{S,\beta})$	
Return 1 if $\beta = \beta'$ AND $\mathcal{Q}_{\mathsf{sk}} \cap S = \phi$.	

Fig. 5. Message Indistinguishability from Unprivileged Users for BED

4 Inadequacy of Existing Schemes

The introduction of this paper claimed all the existing schemes are insufficient. Given the formal definition and security models in the last section, we justify our above claim in this section. For the sake of completeness, we first recall the constructions of all the existing schemes [AD16,AD17,KLEL18]. This is then followed by critical discussions of the proofs of the respective papers. More precisely, we then analyze the security arguments for *group privacy* and the *maximum user accountability* of all the papers and argue that the proofs are incorrect. We found concrete attacks on the *group privacy* of all those constructions. We conclude this section with descriptions of the attacks in detail.

4.1 Overview of Existing Works

We describe all the constructions briefly as per our discussion above. All the papers [AD16,AD17,KLEL18] did not follow a consistent notation. In the following, we bring them into a common notation that we will follow throughout the paper. Firstly, recall that all the existing papers [AD16,AD17,KLEL18] instantiates in the bilinear pairing group system. Therefore, we assume existence of $\mathcal{BG} = (p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{\$} \mathsf{BGen}$ be a prime order symmetric bilinear pairing group system throughout this section, where $\mathbb{G}, \mathbb{G}_T$ are groups of prime order p and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is the bilinear mapping. Let n denote the maximal number of receivers. Let $\mathsf{ID} = \{ID_1, \ldots, ID_n\}$ be the set of identifiers where $ID_i \in \mathbb{Z}^+$ and λ is the security parameter. The privileged user set $S \subseteq \mathsf{ID}$ is of size k'. Let $k \leq n$ be the maximum allowed size of S. Let R be the revoked user set and v is maximum number of revocation possible. We have omitted the decryption function from the description as that is not necessary for our discussion. For a more detailed description, readers are recommended to take a look at [AD16,AD17,KLEL18].

4.1.1 Brief Description of BED Scheme of [AD16] Authors of [AD16] presented their construction from a key encapsulation mechanism with dealership (KEMD). The construction is as follows.

- $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{KEMD.Setup}(1^\lambda, n)$: Generate the public and private key as follows,

1. Let g, h be generators of the group $\mathbb{G}$ and let $H : \{0,1\}^* \to \mathbb{Z}_p^*$ be a cryptographically secure hash function.
2. Sample $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and set master key $\mathsf{msk} = (\alpha, h)$ and publish

$$\mathsf{pk} = \Big(\mathcal{BG}, g, g^{\alpha}, \ldots, g^{\alpha^n}, e(g,h), h^{\alpha}, H, \mathsf{ID}\Big).$$

- $(\mathsf{sk}_i) \leftarrow \mathsf{KEMD.KeyGen}(\mathsf{pk}, \mathsf{msk}, i)$: Set user secret key as

$$\mathsf{sk}_i = h^{\frac{1}{\alpha + H(ID_i)}}$$

and send it to user i via a secure channel.
- $(\mathit{\Gamma}_S, k) \leftarrow \mathsf{KEMD.GroupGen}(\mathsf{pk}, S)$: For a group of users $S = \{ID_1, ID_2, \ldots, ID_{k'}\} \subseteq [n]$, generate the group token $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4)$ as,
  1. Define $P_S(x) = \prod\limits_{ID_j \in S} (x + H(ID_j)) = \sum\limits_{i=0}^{k'} P_i x^i$. P_i's are functions of $H(ID_j)$ for $ID_j \in S$.
  2. Sample $s \xleftarrow{\$} \mathbb{Z}_p$ and generate $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4)$ as,
  $$\begin{aligned} \omega_1 &= h^{-s\alpha} & \omega_3 &= g^{sP_S(\alpha)} \\ \omega_2 &= g_{n-k}^{sP_S(\alpha)} & \omega_4 &= e(g,h)^s. \end{aligned}$$
  3. Set a group threshold k for group size S where $k \geq k' = |S|$.
  4. Send S to users and publish $(\mathit{\Gamma}_S, k)$.
- $(0 \vee 1) \leftarrow \mathsf{KEMD.Verify}(\mathit{\Gamma}_S, \mathsf{pk}, k)$: The verification work as following,

$$\mathsf{KEMD.Verify}(\mathit{\Gamma}_S, \mathsf{pk}, k) = \begin{cases} 1, \text{ if } e(\omega_2, g^{\alpha^k}) = e(\omega_3, g^{\alpha^n}) \\ 0, \text{ otherwise.} \end{cases}$$

- $(\mathsf{Hdr}, K) \leftarrow \mathsf{KEMD.Encrypt}(\mathsf{\Gamma_S}, \mathsf{pk})$: Extract $(\omega_1, \omega_3, \omega_4)$ from $\mathit{\Gamma}_S$, sample $r \xleftarrow{\$} \mathbb{Z}_p$ and set $K = \omega_4^r$ and $\mathsf{Hdr} = (C_1, C_2) = (\omega_1^r, \omega_3^r)$, and then publish Hdr and keep K secret.

4.1.2 Brief Description of BED Scheme of [AD17] This follow-up work by the same authors did not use a KEMD. The BED construction of [AD17] is the following.

- $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{BED.Setup}(n, 1^{\lambda})$:
  1. Sample $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and set,
  $$\mathsf{pk} = (\mathcal{BG}, l_0, l_0^{\alpha}, \ldots, l_0^{\alpha^n}, g, g^{\alpha}, \ldots, g^{\alpha^{n+1}}, e(g,g), e(g,l_0), \mathsf{ID}); \quad \mathsf{msk} = (\alpha),$$
  where g is generator of $\mathbb{G}$ and l_0 is a random non-identity element of $\mathbb{G}$.
  2. Keep msk secret and publish pk.
- $(\mathsf{sk}_i) \leftarrow \mathsf{BED.KeyGen}(\mathsf{pk}, \mathsf{msk}, i)$: Sample $h_i \xleftarrow{\$} \mathbb{G}$ and $r_i \xleftarrow{\$} \mathbb{Z}_p$ and generate $\mathsf{sk}_i = (d_{1,i}, d_{2,i}, d_{3,i}, \mathsf{label}_i)$ as,
$$\begin{aligned} d_{1,i} &= (h_i \cdot g^{r_i})^{\frac{1}{\alpha(\alpha + ID_i)}} & d_{2,i} &= r_i \\ d_{3,i} &= \Big(h_i \cdot l_0^{d_{2,i}}\Big)^{\frac{1}{\alpha}} & \mathsf{label}_i &= \Big(h_i, h_i^{\alpha} \ldots, h_i^{\alpha^n}\Big). \end{aligned}$$
Send sk_i to user i through a secure channel.
- $(\mathit{\Gamma}_S, k) \leftarrow \mathsf{BED.GroupGen}(\mathsf{pk}, S)$: Select a threshold value k and a group $S = \{ID_{i_1}, \ldots, ID_{i_{k'}}\} \subseteq [n]$ of k' many users where $k' \leq k$ and generate group token $(\mathit{\Gamma}_S)$ as following,
  1. Define $P_S(x) := \prod\limits_{ID_{i_j} \in S} (x + ID_{i_j})$.
  2. Sample $s \xleftarrow{\$} \mathbb{Z}_p$ and generate group token $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ as following,
  $$\begin{aligned} \omega_1 &= g^{s\alpha P_S(\alpha)} & \omega_2 &= g^{s\alpha^{n-k+1} P_S(\alpha)} & & \\ \omega_4 &= e(g,g)^{-s} & \omega_5 &= e(g,l_0)^s & \omega_3 &= g^{-s\alpha}. \end{aligned}$$

Send S to subscribed users via a secure channel and publish $(\mathit{\Gamma}_S, k)$.

- $(0 \vee 1) \leftarrow \mathsf{BED.Verify}(\mathit{\Gamma}_S, \mathsf{pk}, k)$: Parse $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ and checks,
$$\mathsf{BED.Verify}(\mathit{\Gamma}_S, \mathsf{pk}, k) = \begin{cases} 1, \text{ if } e(\omega_1, g^{\alpha^n}) = e(\omega_2, g^{\alpha^k}) \\ 0, \text{ otherwise.} \end{cases}$$
- $(\mathsf{ct}) \leftarrow \mathsf{BED.Encrypt}(\mathit{\Gamma}_S, \mathsf{pk}, M)$: Parse $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$, sample $r \xleftarrow{\$} \mathbb{Z}_p$ and compute ciphertext as
$$\mathsf{ct} = (ct_1, ct_2, ct_3, ct_4) = ((\omega_1^r, \omega_3^r, \omega_4^r, M \cdot \omega_5^r)).$$

4.1.3 Brief Description of BED Scheme of [KLEL18] The construction of [KLEL18] the acronym RR was used for recipient revocation. They referred to their scheme as RR-BED.

- $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{RR\text{-}BED.Setup}(1^\lambda, n)$: Choose $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p$, $h \xleftarrow{\$} \mathbb{G}$ and compute,
$$\begin{aligned} \mathsf{pk} =& (\mathcal{BG}, h, h^\alpha, \ldots, h^{\alpha^n}, g, g^\alpha, \ldots, g^{\alpha^n}, \\ & g^{\alpha\beta}, \ldots, g^{\alpha^{n+1}\beta}, e(g,g), e(g,h), \mathsf{ID}) \\ \mathsf{msk} =& (\alpha, \beta). \end{aligned}$$
Keep msk secret and publish pk.
- $(\mathsf{sk}_i) \leftarrow \mathsf{RR\text{-}BED.KeyGen}(\mathsf{pk}, \mathsf{msk}, i)$: Sample $l_i \xleftarrow{\$} \mathbb{G}$ and $r_i \xleftarrow{\$} \mathbb{Z}_p$ and generate $\mathsf{sk}_i = (d_{1,i}, d_{2,i}, d_{3,i}, \mathsf{label}_i)$ as,
$$d_{1,i} = (l_i \cdot g^{r_i})^{\frac{1}{\alpha\beta(\alpha + ID_i)}} \qquad d_{2,i} = r_i$$
$$d_{3,i} = \left(l_i h_i^{d_{2,i}}\right)^{\frac{1}{\alpha\beta}} \qquad \mathsf{label}_i = \left(l_i, l_i^\alpha \ldots, l_i^{\alpha^n}\right).$$
Send sk_i to user i through a secure channel.
- $(\mathit{\Gamma}_S) \leftarrow \mathsf{RR\text{-}BED.GroupGen}(\mathsf{pk}, S, k, v)$: Select a threshold value k and a group $S = \{ID_{i_1}, \ldots, ID_{i_{k'}}\} \subseteq [n]$ of k' many users where $k' \leq k$ and generate group token $(\mathit{\Gamma}_S)$ as following,
  1. Define $P_S(x) := \prod_{ID_{i_j} \in S} (x + ID_{i_j})$.
  2. Sample $s \xleftarrow{\$} \mathbb{Z}_p$ and set $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ as,
$$\omega_1 = g^{s\alpha\beta P_S(\alpha)} \qquad \omega_2 = g^{s\alpha^{n-v+1}\beta P_S(\alpha)}$$
$$\omega_4 = e(g,g)^{-s} \qquad \omega_5 = e(g,h)^s \qquad \omega_3 = [\widehat{\omega}_i] = [g^{-s\alpha^i}]_{1 \leq i \leq k+1}.$$

  Send S to subscribed users via a secure channel and publish $(\mathit{\Gamma}_S, k)$.
- $(0 \vee 1) \leftarrow \mathsf{RR\text{-}BED.Verify}(\mathit{\Gamma}_S, \mathsf{pk}, k)$: Parse $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ and check,
$$\mathsf{BED.Verify}(\mathit{\Gamma}_S, \mathsf{pk}, k) = \begin{cases} 1, \text{ if } e(\omega_1, g^{\alpha^n}) = e(\omega_2, g^{\alpha^k}) \\ 0, \text{ otherwise.} \end{cases}$$
- $(\mathsf{ct}) \leftarrow \mathsf{RR\text{-}BED.Encrypt}(\mathit{\Gamma}_S, \mathsf{pk}, M)$: Parse $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$, sample $r \xleftarrow{\$} \mathbb{Z}_p$ and compute ciphertext as
$$\mathsf{ct} = (ct_1, ct_2, \widehat{ct_1}, \ldots, \widehat{ct_{k+1}}, ct_4) = ((\omega_1^r, \omega_3^r, \widehat{\omega_1}^r, \ldots, , \widehat{\omega_{k+1}}^r, M \cdot \omega_5^r)).$$
- $(\mathsf{ct}') \leftarrow \mathsf{RR\text{-}BED.Revoke}(\mathsf{ct}, R, \mathsf{pk})$: Parse ct as $(ct_1, ct_2, \widehat{ct_1}, \ldots, \widehat{ct_{k+1}}, ct_4)$. Let $R = \{ID_{i_1}, \ldots, ID_{i_l}\} \subseteq S$ where $l \leq v$. Generate $\mathsf{ct}' = (ct'_1, ct'_2, \widehat{ct}'_1, ct'_3)$ as,
  1. If $R = \phi$, $\mathsf{ct}' = (ct'_1, ct'_2, \widehat{ct}'_1, ct'_3) = (ct_1, ct_2, \widehat{ct_1}, ct_4)$.
  2. If $R \neq \phi$, it compute $\frac{\prod_{ID_j \in R} (x + ID_j)}{\prod_{ID_j \in R} (ID_j)} = \sum_{i=0}^{l} f_i \alpha^i$ where $f_0 = 1.$ and $H = \prod_{i=2}^{l} \widehat{ct}_i^{f_i} = g^{-t \sum_{i=2}^{l} f_i \alpha^i}$. Set
$$y = t \sum_{i=0}^{l} f_i \alpha^i \text{ where } t = rs \text{ (the random coin chosen by dealer and broadcaster) and compute,}$$

$$\begin{aligned} ct_1 &= g^{\alpha\beta y\left(\prod_{ID_j \in G}(\alpha+ID_i)\right)} & \qquad ct_2 &= e(g,g)^{-y} \\ \widehat{ct}_1 &= g^{-\alpha y} & \qquad ct_3 &= M \cdot e(g,h)^{y}. \end{aligned}$$

4.2 Flaws in Security Argument of Previous Works

All three constructions [AD16,AD17,KLEL18] followed a similar path to argue the security of *group privacy* and *maximum user accountability.* To keep our presentation concise and straightforward, we discuss the flaw in the security argument of [AD16]. We omit the discussion of flaws in [AD17,KLEL18] as the flaws are exactly the same as in [AD16].

4.2.1 Issue with Group Privacy Recall from Section 3.3.1, the *group privacy* security game of BED allows an adversary to submit two privileged sets of the same size S_0 and S_1. The challenger chooses a bit $b \xleftarrow{\$} \{0,1\}$ and returns a group token for S_b. The adversary wins if it correctly guesses b. All three constructions [AD16,AD17,KLEL18] argued that any adversary could predict b if and only if it can compute the randomness used in the challenge group token. They also argued that computing the group token $\mathit{\Gamma}_{S_b}$ by any adversary is the equivalent to computing the randomness s from ω^{-s} where it knows ω from the public parameter. This is equivalent to solving the discrete logarithm problem in $\mathbb{G}$.

We think this argument of security is incorrect. If one follows the argument above carefully, [AD16] argued that if an adversary can break the discrete logarithm problems, then we can also break their scheme. In provable security, we usually do the opposite. Precisely, the argument should be the converse, i.e., if an adversary can break the *group privacy* security of [AD16] then we can also break the discrete logarithm problem.

Looking ahead, it is, in fact, easy to check [AD16] is not secure. We have described a concrete attack on [AD16] in the security settings of *group privacy* in Section 4.3.1. The attack idea is simply the public parameters, and the group token forms a DDH instance for any adversarially chosen $S \subseteq [n]$. This attack also works on the later works [AD17,KLEL18] as we will see in Section 4.3.

4.2.2 Issue with Maximum User Accountability Recall, we denote S as the privileged set and k be the maximum allowed size for S. Maximum user accountability security ensures that a group token $\mathit{\Gamma}_S$ can not pass the verification stage if $|S| > k$. To argue this security, existing works [AD16,AD17,KLEL18] used the standard (f,n)-DHE assumption defined in [GSP$^+$16]. Informally, the (f,n)-DHE problem is as following,

> *The (f,n)-Diffie-Hellman Exponent Assumption*: Says, given an instance $\left(\mathcal{G}, g, g^{\alpha}, \ldots, g^{\alpha^n}\right)$ where $\mathcal{G} = (e, \mathbb{G}, \mathbb{G}_T, p) \xleftarrow{\$} \mathsf{BGen}$ is a symmetric bilinear pairing group system and g be a random generator of $\mathbb{G}$ and $\alpha \xleftarrow{\$} \mathbb{Z}_p$. The problem is to find a pair $(f(x), g^{f(\alpha)})$ where $f(x)$ is a polynomial of degree $n' > n$.

We discuss the security flaw in the security proof of [AD16] and reiterate that the other two constructions [AD17,KLEL18] also use a similar argument. To explain the problem in the proof of [AD16], we present a gist of their proof. Let $\mathcal{A}$, an adversary, trying to break the *maximum user accountability* of [AD16]. Let $\mathcal{B}$ be another adversary trying to solve the (f,n)-DHE problem. Adversary $\mathcal{B}$ uses $\mathcal{A}$ as a subroutine. Given the (f,n)-DHE problem instance, $\mathcal{B}$ simulates the required public parameters for $\mathcal{A}$ (description of simulation is not required here). $\mathcal{B}$ also submits challenge value k as the maximum size of the privileged set. $\mathcal{A}$ compute a privileged set S^* such that $|S^*| = k' > k$ and generate the group token,

$$\mathit{\Gamma}_{S^*} = (\widehat{\omega}_1, \widehat{\omega}_2, \widehat{\omega}_3, \widehat{\omega}_4) = \left(h^{-s\alpha}, g^{s\alpha^{n-k}\widehat{P}(\alpha)}, g^{s\widehat{P}(\alpha)}, e(g,h)^s\right)$$

where $s \in \mathbb{Z}_p$, $\widehat{P}_{S^*}(x) = \prod_{ID \in S^*}(x + H(ID))$. $\mathcal{A}$ sends $(\mathit{\Gamma}_{S^*}, S^*)$ to $\mathcal{B}$ [5]. Now the argument they provided was, set $f(x) = sx^{n-k}\widehat{P}_{S^*}(x)$. Notice $\widehat{P}_{S^*}(x)$ is a polynomial of degree k'. So, $f(x)$ is a polynomial of degree $n - k + k' > n$ as $k' > k$. So they claimed $\left((f(x), \widehat{\omega}_2 = g^{f(\alpha)}\right)$ is the (f,N)-DHE solution for $\mathcal{B}$.

[5] Note that, BED does not allow dealer to send the description of privileged set S^* for which the token is generated.

The following section discusses the inconsistencies in their security arguments,

- Note that $\mathcal{A}$ can generate a valid $g^{\widehat{P}_{S^*}(\alpha)}$ from the pk as long as $|S^*| = k' \leq n$. $\mathcal{A}$ samples $s \xleftarrow{\$} \mathbb{Z}_p$ and computes $\varGamma_{S^*}$. But the problem is $\mathcal{A}$ which models the dealer in a BED system can not submit s used in generating $\varGamma_{S^*}$. So, $\mathcal{B}$ can not compute $f(x) = sx^{n-k}\widehat{P}_{S^*}(x)$.
- Also observe that, $\mathcal{A}$ in all the security arguments of [AD16,AD17,KLEL18] submits $(\varGamma_{S^*}, S^*)$ as a challenge. Submitting S^* is not consistent/permitted in the security definition of *maximum user accountability* in Section 3.3.2. In fact, the group privacy security model (Section 3.3.1) dictates that $\mathcal{A}$ which models the dealer here, should not reveal the set description of S^*. Also giving $(\varGamma_{S^*}, S^*)$ together does not have any significance.

Now if we consider $\mathcal{A}$ does not submit S^* (the privileged user set) and/or s (the randomness) with it's challenge, then the security proof does not hold. As $\mathcal{B}$ can not construct $\widehat{P}_{S^*}(x)$ as well as $f(x)$ without knowing S^* and/or s. Hence the simulation they provided is incorrect.

4.3 Attacks on the Group Privacy of Existing Works

4.3.1 Description of Attack on [AD16] We are all set to show [AD16] is not secure in the *"group privacy"* security model as described in Section 3.3.1.

Remember from Section 4.1.1, the public parameter of [AD16] contains $\mathsf{pk} = \big(\mathcal{G}, g, g^{\alpha}, \ldots, g^{\alpha^n}, h^{\alpha}, H, \mathsf{ID}\big)$ where g and h are random generator of $\mathbb{G}$. Let adversary choose any $S \subseteq [n]$. The adversary can easily compute $g^{P_S(\alpha)}$ using pk as $P_S(x) = \prod_{ID_j \in S} (x + H(ID_j))$. Also recall, group token $\varGamma_S$ for set S contains $\varGamma_S = (\omega_1, \omega_2, \omega_3, \omega_4)$ where,

$$\omega_1 = h^{-s\alpha} \qquad \omega_3 = g^{sP_S(\alpha)}$$

$$\omega_2 = g_{n-k}^{sP_S(\alpha)} \qquad \omega_4 = e(g,h)^{s}.$$

The security game in Section 3.3.1 allows an adversary to choose any two sets S_0 and S_1 of size $k' \leq n$. Challenger generates group token for S_b where $b \xleftarrow{\$} \{0,1\}$, to which adversary has to guess b.

Note that, $g^{P_{S_b}(\alpha)}$ and $\varGamma_{S_b}$ together forms a DDH tuple and therefore evaluating symmetric bilinear pairing breaks the group privacy security for $b \in \{0,1\}$. The concrete attack is as follows.

Assume the adversary has chosen any $S_0, S_1 \subseteq [n]$ with $|S_0| = |S_1| = k' \leq n$. The challenger chooses a s and a random bit $b \xleftarrow{\$} \{0,1\}$ and publishes the group token $(\varGamma_{S_b}, k)$, where $k' \leq k$. Adversary computes $g^{P_{S_0}(\alpha)}$ using pk. Adversary can do so as $k' \leq n$. Now the adversary evaluate the following pairings,

$$\left(g^{P_{S_0}(\alpha)}, \omega_1\right) = e\,(g,h)^{-s\alpha P_{S_0}(\alpha)}$$

$$e\,(\omega_3, h^{\alpha}) = e(g,h)^{s\alpha P_{S_b}(\alpha)}$$

From the above two pairings adversary can easily compute b as,

$$b = \begin{cases} 0 & \text{if } e(g^{P_{S_0}(\alpha)}, \omega_1) \times e(\omega_3, h^{\alpha}) = 1 \\ 1 & \text{otherwise.} \end{cases} \qquad (1)$$

Note that, if the random bit b chosen by the challenger is 0, $e(\omega_3, h^{\alpha})$ would indeed be $e(g,h)^{s\alpha P_{S_0}(\alpha)}$, and the product $e\left(g^{P_{S_0}(\alpha)}, \omega_1\right) \times e(\omega_3, h^{\alpha})$ would be 1. Thus the adversary guesses b correctly with overwhelming probability.

4.3.2 Description of attack on [AD17] The construction of [AD17] suffer from similar vulnerability. Precisely, the pk in [AD17] contains $(g, g^{\alpha}, \ldots, g^{\alpha^n})$. Here also adversary can easily compute $g^{P_S(\alpha)}$ using pk as $P_S(x) = \prod_{ID_{i_j} \in S} (x + ID_{i_j})$. Description of group token in this case is $\varGamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ as following,

$$\begin{array}{lll} \omega_1 = g^{s\alpha P_S(\alpha)} & \omega_2 = g^{s\alpha^{n-k+1} P_S(\alpha)} & \\ & & \omega_3 = g^{-s\alpha}. \\ \omega_4 = e(g,g)^{-s} & \omega_5 = e(g, l_0)^s & \end{array}$$

Proceeding similar to the previous attack, the adversary chooses two privileged sets of equal size, S_0, S_1 and submits those to challenger. The challenger chooses $b \xleftarrow{\$} \{0,1\}$ and returns group token for S_b. The group token S_b contains $\omega_1 = g^{-s\alpha}$ and $\omega_3 = g^{sP_{S_b}(\alpha)}$. The adversary generates $g^{P_{S_0}(\alpha)}$ from the public parameters. Adversary proceeds by computing the following pairings $e\left(g^{P_{S_0}(\alpha)}, \omega_3\right)$ and $e(\omega_1, g)$.

$$e\left(g^{P_{S_0}(\alpha)}, \omega_3\right) = e(g,g)^{-s\alpha P_{S_0}(\alpha)}$$
$$e(\omega_1, g) = e(g,g)^{s\alpha P_{S_b}(\alpha)}$$

With this adversary guesses b as,

$$b = \begin{cases} 0 & \text{if } e\left(g^{P_{S_0}(\alpha)}, \omega_3\right) \times e(\omega_1, g) = 1 \\ 1 & \text{otherwise.} \end{cases} . \tag{2}$$

That is, if the random bit b chosen by the challenger is 0, then the product $e\left(g^{P_{S_0}(\alpha)}, \omega_3\right) \times e(\omega_1, g)$ indeed be 1. Thus the adversary can guess b correctly with overwhelming probability.

4.3.3 Description of Attack on [KLEL18] This construction uses a similar kind of idea used in [AD17] to hide the set information in the group token. So a similar type of attack is possible for this construction as well. The public parameters in [KLEL18] contains
$\mathsf{pk} = (\mathcal{G}, h, h^{\alpha}, \ldots, h^{\alpha^n}, g, g^{\alpha}, \ldots, g^{\alpha^n}, g^{\alpha\beta}, \ldots, g^{\alpha^{n+1}\beta}, e(g,g), e(g,h), \mathsf{ID})$. Like previous two constructions here also adversary can compute $g^{\alpha\beta P_{S_b}(\alpha)}$ as $P_S(x) = \prod_{ID_{i_j} \in S} (x + ID_{i_j})$. Group token in their construction $\varGamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ as following,

$$\begin{array}{lll} \omega_1 = g^{s\alpha\beta P_S(\alpha)} & \omega_2 = g^{s\alpha^{n-v+1}\beta P_S(\alpha)} & \\ \omega_4 = e(g,g)^{-s} & \omega_5 = e(g,h)^s & \omega_3 = [\widehat{\omega}_i] = [g^{-s\alpha^i}]_{1 \le i \le k+1}. \end{array}$$

Our attack here would be, the adversary generates $g^{\alpha\beta P_{S_0}(\alpha)}$ from the public parameters. Adversary proceeds by computing the pairings $e\left(g^{\alpha\beta P_{S_0}(\alpha)}, \widehat{\omega_1}\right)$ and $e(\omega_1, g^{\alpha})$.

$$e\left(g^{\alpha\beta P_{S_0}(\alpha)}, \widehat{\omega_1}\right) = e(g,g)^{-s\alpha^2\beta P_{S_0}(\alpha)}$$
$$e(\omega_1, g^{\alpha}) = e(g,g)^{s\alpha^2\beta P_{S_b}(\alpha)}$$

With this adversary guesses b as,

$$b = \begin{cases} 0 & \text{if } e\left(g^{\alpha\beta P_{S_0}(\alpha)}, \widehat{\omega_1}\right) \times e(\omega_1, g^{\alpha}) = 1 \\ 1 & \text{otherwise.} \end{cases} . \tag{3}$$

Here also, if the random bit b chosen by the challenger is 0, then the product $e\left(g^{\alpha\beta P_{S_0}(\alpha)}, \widehat{\omega_1}\right) \times e(\omega_1, g^{\alpha})$ indeed be 1. Thus the adversary can guess b correctly with overwhelming probability.

5 BED with Constant-size Ciphertext and Key ($\mathsf{bed}_{\mathcal{C}}$)

Here we present our first broadcast encryption with dealership ($\mathsf{bed}_{\mathcal{C}}$) construction. Recall that, in a BED, a message is encrypted for a set of users. Our first BED construction here achieves constant-size ciphertext i.e., the ciphertext size is independent of the size of the set. This construction is heavily influenced by the broadcast encryption of [GLR18]. Note that, the broadcast encryption of [GLR18] does not have anonymous security. Our novelty in this construction is primarily that we still manage to get the group privacy security (priv).

5.1 Construction

It is defined by the following ppt algorithms.

- $\mathsf{Setup}(1^\lambda, n)$:
  1. $\mathcal{CG} = (p, q, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, g_p, g_q, h_p, h_q, e) \leftarrow \mathsf{CBGen}(1^\lambda)$ where $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$ are cyclic groups of order $N = pq$ and e is an admissible type-3 bilinear map. $\mathbb{G}_r$ (resp. $\mathbb{H}_r$) is a subgroup of $\mathbb{G}$ (resp. $\mathbb{H}$) of order r. Further, $e(g_p, h_q) = e(g_q, h_p) = 1$.
  2. Sample $\alpha, \gamma \xleftarrow{\$} \mathbb{Z}_N$.
  3. Set $\mathsf{msk} = (\alpha, \gamma, h_p, g_q, h_q)$.
  4. Set $\mathsf{pk} = (g_p, g_p^\alpha, \ldots, g_p^{\alpha^n}, g_p^\gamma, u_1 = h_p^\alpha, \ldots, u_n = h_p^{\alpha^n}, e(g_p, h_p)^\gamma, \mathsf{H})$ where $\mathsf{H} : \mathbb{G}_T \to \mathcal{K}$ is a universal hash function for encapsulation key space $\mathcal{K} = \{0,1\}^m$ where $m = \mathsf{poly}(\lambda)$.
- $\mathsf{KeyGen}(\mathsf{msk}, x_i)$: Set $\mathsf{sk}_{x_i} = h_p^{\frac{\gamma}{(\alpha + x_i)}}$.
- $\mathsf{GroupGen}(\mathsf{pk}, S, k)$:
  1. Suppose $S = \{ID_1, \ldots, ID_{k'}\}$ where $k' \leq k \leq n$.
  2. Set $P_S(z) = \prod_{y \in S} (z + y) = b_0 + b_1 z + \ldots + b_k z^{k'}$ where $|S| = k'$.
  3. Sample $s \xleftarrow{\$} \mathbb{Z}_N$.
  4. Output $(k, \mathit{\Gamma}_S) = (k, \omega_1, \omega_2, \omega_3, \omega_4)$ where
  $$\begin{aligned} \omega_1 &= g_p^{\gamma s} & \omega_2 &= g_p^{s P_S(\alpha)} \\ \omega_3 &= g_p^{\alpha^{n-k} s P_S(\alpha)} & \omega_4 &= e(g_p, h_p)^{\gamma s}. \end{aligned}$$
- $\mathsf{Verify}(\mathsf{pk}, \mathit{\Gamma}_S, k)$: If $e(\omega_2, h_p^{\alpha^n}) = e(\omega_3, h_p^{\alpha^k})$, output 1 else 0.
- $\mathsf{Enc}(\mathsf{pk}, \mathit{\Gamma}_S, M)$:
  1. Sample $r \xleftarrow{\$} \mathbb{Z}_N$.
  2. Output $\mathsf{CT}_S = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_0) = (\omega_1^r, \omega_2^r, M \oplus \mathsf{H}(\omega_4^r))$
- $\mathsf{Dec}(\mathsf{pk}, (\mathsf{sk}, x), (\mathsf{CT}_S, S))$:
  1. Parse $\mathsf{CT}_S = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_0) = (g_p^{\gamma t}, g_p^{t P_S(\alpha)}, M \oplus \mathsf{H}(e(g_p, h_p)^{\gamma t}))$.
  2. Compute $P_{S \setminus \{x\}}(z) = \prod_{y \in S \setminus \{x\}} (z + y) = a_0 + a_1 z + \ldots + a_{k-1} z^{k-1}$.
  3. Compute $A = e(\mathsf{ct}_2, \mathsf{sk}) \cdot e\left(\mathsf{ct}_1, \prod_{i \in [1,k]} h_p^{a_i \alpha^i}\right)^{-1}$.
  4. Output $\mathsf{ct}_0 \oplus \mathsf{H}(A^{a_0^{-1}})$.

Correctness. $e(\mathsf{ct}_2, \mathsf{sk}) = e(g_p^{t P_S(\alpha)}, h_p^{\frac{\gamma}{(\alpha + x)}}) = e(g_p, h_p)^{\gamma t P_{S \setminus \{x\}}(\alpha)}$.

$e(\mathsf{ct}_1, \prod_{i \in [1,k]} h_p^{a_i \alpha^i}) = e(g_p^{\gamma t}, h_p^{P_{S \setminus \{x\}}(\alpha) - a_0}) = e(g_p, h_p)^{\gamma t P_{S \setminus \{x\}}(\alpha)} e(g_p, h_p)^{-a_0 \gamma t}$.

Thus, $\mathsf{H}(A^{a_0^{-1}}) = \mathsf{H}(e(g_p, h_p)^{\gamma t})$ and the correctness holds naturally.

5.2 Security

We prove the above construction achieves security in all four security models.

5.2.1 Group Privacy

Theorem 1. *Let there exist a* ppt *adversary* $\mathcal{A}$ *breaking the group privacy of* $\mathsf{bed}_{\mathcal{C}}$ *with a non-negligible advantage. Then, there is a* ppt *adversary* $\mathcal{B}$ *which has a non-negligible advantage in solving the Decisional Diffie-Hellman problem in* $\mathbb{G}_p$ *where* $\mathcal{CG} = (p, q, g_p, g_q, h_p, h_q, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) \leftarrow \mathsf{CBGen}(1^\lambda)$ *such that* $\mathbb{G}_p = \langle g_p \rangle$*.*

Proof. We are given a ppt adversary $\mathcal{A}$ of priv security, and we want to construct a ppt adversary $\mathcal{B}$ for DDH that uses $\mathcal{A}$ as a subroutine. Given a DDH_C problem instance $(g_p, g_q, h_p, h_q, g_p^a, g_p^{ab}, Z)$ for $a, b \overset{\$}{\leftarrow} \mathbb{Z}_p$, $\mathcal{B}$ does the following:

- Sample $\alpha \overset{\$}{\leftarrow} \mathbb{Z}_p$.
- Implicitly set $\gamma = a$ and $s = b$.
- Publish the public key $\mathsf{pk} = (g_p, g_p^\alpha, \dots, g_p^{\alpha^n}, g_p^a, h_p^\alpha, \dots, h_p^{\alpha^n}, e(g_p, h_p)^\gamma, \mathsf{H})$.
- Given pk, $\mathcal{A}$ outputs two sets (S_0, S_1) such that $|S_0| = |S_1| = k' < n$.
- $\mathcal{B}$ then chooses $\beta \overset{\$}{\leftarrow} \{0, 1\}$, and return the token $\mathit{\Gamma}_{S_\beta} = (\omega_1, \omega_2, \omega_3, \omega_4)$ as following:
$$\begin{aligned} \omega_1 &= g_p^{ab} & \omega_2 &= Z^{P_{S_\beta}(\alpha)} \\ \omega_3 &= Z^{\alpha^{n-k} P_{S_\beta}(\alpha)} & \omega_4 &= e(\omega_1, h_p) \end{aligned}.$$
- $\mathcal{A}$ outputs β'.
- If $\beta' = \beta$, then $\mathcal{B}$ outputs 1 else 0.

As α is chosen uniformly at random, the public key pk is properly distributed. If $Z = g_p^b$ then $\mathit{\Gamma}_{S_\beta}$ is a valid group token to $\mathcal{A}$, and if Z is chosen uniformly random then both ω_2 and ω_3 are just two random elements of $\mathbb{G}_p$.

Then, the advantage of $\mathcal{B}$ in the DDH game is same as the adversary $\mathcal{A}$ guessing β with probability anything other than guessing randomly. So,

$$\begin{aligned}
\Pr\left[\mathcal{B}^{\mathsf{DDH}_{\mathcal{C}}} \Rightarrow 1\right] &\geq \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_{\mathcal{C}}} \Rightarrow 1\right] \\
\Pr\left[\mathcal{B}^{\mathsf{DDH}_{\mathcal{C}}} \Rightarrow 1\right] &\geq \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_{\mathcal{C}}} \Rightarrow 1 | Z = g_p^b\right] \Pr\left[Z = g_p^b\right] \\
&\quad + \sum_{p-1} \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_{\mathcal{C}}} \Rightarrow 1 | Z \overset{\$}{\leftarrow} \mathbb{G}_p\right] \Pr\left[Z \overset{\$}{\leftarrow} \mathbb{G}_p\right] \\
\frac{1}{p} \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_{\mathcal{C}}} \Rightarrow 1 | Z = g_p^b\right] &\leq \left| \Pr\left[\mathcal{B}^{\mathsf{DDH}_{\mathcal{C}}} \Rightarrow 1\right] - \frac{p-1}{p} \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_{\mathcal{C}}} \Rightarrow 1 | Z \overset{\$}{\leftarrow} \mathbb{G}_p\right] \right| \\
\frac{1}{p} \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_{\mathcal{C}}} \Rightarrow 1 | Z = g_p^b\right] &\leq \left| \Pr\left[\mathcal{B}^{\mathsf{DDH}_{\mathcal{C}}} \Rightarrow 1\right] - \frac{p-1}{2p} \right| \\
\frac{1}{p} \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_{\mathcal{C}}} \Rightarrow 1 | Z = g_p^b\right] &\leq \Big| \Pr\left[\mathcal{B}^{\mathsf{DDH}_{\mathcal{C}}} \Rightarrow 1 | Z = g_p^b\right] \Pr\left[Z = g_p^b\right] \\
&\quad + \sum_{p-1} \Pr\left[\mathcal{B}^{\mathsf{DDH}_{\mathcal{C}}} \Rightarrow 1 | Z \overset{\$}{\leftarrow} \mathbb{G}_p\right] \Pr\left[Z \overset{\$}{\leftarrow} \mathbb{G}_p\right] - \frac{p-1}{2p} \Bigg| \\
\Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_{\mathcal{C}}} \Rightarrow 1 | Z = g_p^b\right] &\leq \left| \Pr\left[\mathcal{B}^{\mathsf{DDH}_{\mathcal{C}}} \Rightarrow 1 | Z = g_p^b\right] + (p-1)\left(\frac{1}{2} - \frac{1}{2}\right) \right| \\
\mathsf{Adv}^{\mathsf{priv}}_{\mathcal{A}, \mathsf{bed}_{\mathcal{C}}} &\leq \mathsf{Adv}^{\mathsf{DDH}_{\mathcal{C}}}_{\mathcal{B}}
\end{aligned}$$

□

5.2.2 Maximum Users Accountability Let us consider three random encoding function $\sigma_G : \mathbb{G} \to \{0, 1\}^{m_G}$, $\sigma_H : \mathbb{H} \to \{0, 1\}^{m_H}$ and $\sigma_T : \mathbb{G}_T \to \{0, 1\}^{m_T}$ w.l.o.g. $m_G \leq m_H \leq m_T$.

Theorem 2. *Let $\mathcal{A}$ be a* ppt *algorithm act as an adversary for the maximum user accountability security of* $\mathsf{bed}_{\mathcal{C}}$ *in the generic group model. Let m be a bound on the total number of group elements $\mathcal{A}$ receives from queries it makes to the oracles computing the group actions in $\mathbb{G}$, $\mathbb{H}$, $\mathbb{G}_T$ and the bilinear map e. Then we have that the advantage of $\mathcal{A}$ in the* maximum user accountability *security game of* $\mathsf{bed}_{\mathcal{C}}$ *is at most $\mathcal{O}(m^2/p)$.*

Proof. Let $\mathcal{C}$ denote an algorithm that simulates the generic bilinear group for $\mathcal{A}$. To answer to oracle queries, $\mathcal{C}$ maintains three lists,

$$\begin{aligned} L_G &= \{(f_{G,i}, \sigma_{G,i}) : i \in [0, \psi_G - 1]\} \\ L_H &= \{(f_{H,i}, \sigma_{H,i}) : i \in [0, \psi_H - 1]\} \\ L_T &= \{(f_{T,i}, \sigma_{T,i}) : i \in [0, \psi_T - 1]\} \end{aligned}$$

such that at each step ψ of the game, the relation $\psi_G + \psi_H + \psi_T = \psi + 2n + 3$ holds. Here $f_{*,*}$ are multivariate polynomials over 4 variables α, γ, r, s and $\sigma_{b,i}$ are strings from $\{0,1\}^{m_b}$, where $b \in \{G, H, T\}$. In the course of this proof we use $b \in \{G, H, T\}$ in the subscript to denote a representative of a general groups if not mentioned otherwise. At the beginning of the game i.e., $\psi = 0$, the lists are initialized by setting $\psi_G = (n+2)$, $\psi_H = n$ and $\psi_T = 1$. The polynomials 1, α, $\ldots$, α^n and γ are assigned to $f_{G,0}$, $f_{G,1}$, $\ldots$, $f_{G,n}$, $f_{G,n+1}$; α, $\ldots$, α^n, are assigned to $f_{H,1}$, $f_{H,2}$, $\ldots$, $f_{H,n}$; and γ is assigned to $f_{T,0}$.

These encodings for these polynomials are strings uniformly chosen from $\{0,1\}^{m_b}$ without repetition for polynomials $f_{b,*}$. We assume that $\mathcal{A}$ queries the oracles on strings previously obtained from $\mathcal{C}$ and naturally $\mathcal{C}$ can obtain the index of a given string $\sigma_{b,i}$ in the list L_b. The oracles are simulated as follows.

Group Actions in $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{G}_T$. We describe this for the group $\mathbb{G}$. We note that the group actions in $\mathbb{H}$ and $\mathbb{G}_T$ are simulated similarly. If $\mathcal{A}$ submits two strings $\sigma_{G,i}$ and $\sigma_{G,j}$ and a sign bit indicating addition or subtraction. $\mathcal{C}$ first finds $f_{G,i}$ and $f_{G,j}$ corresponding to $\sigma_{G,i}$ and $\sigma_{G,j}$ respectively in L_G and computes $f_{G,\psi_G} = f_{G,i} \pm f_{G,j}$. If there exists an index $k \in [0, \psi_G - 1]$, such that $f_{G,\psi_G} = f_{G,k}$, $\mathcal{C}$ sets $\sigma_{G,\psi_G} = \sigma_{G,k}$; otherwise $\mathcal{C}$ sets $\sigma_{G,\psi_G} \xleftarrow{\$} \{0,1\}^{m_G} \setminus \{\sigma_{G,0}, \sigma_{G,1}, \ldots, \sigma_{G,\psi_G - 1}\}$, add $(f_{G,\psi_G}, \sigma_{G,\psi_G})$ to L_G, returns σ_{G,ψ_G} to $\mathcal{A}$ and increment ψ_G by one.

Bilinear Map. If $\mathcal{A}$ submits two strings $\sigma_{G,i}$ and $\sigma_{H,j}$, $\mathcal{C}$ first finds $f_{G,i}$ in L_G corresponding to $\sigma_{G,i}$ and $f_{H,j}$ in L_H corresponding to $\sigma_{H,j}$ respectively and computes $f_{T,\psi_T} = f_{G,i} \pm f_{H,j}$. If there exists an index $k \in [0, \psi_T - 1]$, such that $f_{T,\psi_T} = f_{T,k}$, $\mathcal{C}$ sets $\sigma_{T,\psi_T} = \sigma_{T,k}$; otherwise $\mathcal{C}$ sets $\sigma_{T,\psi_T} \xleftarrow{\$} \{0,1\}^{m_T} \setminus \{\sigma_{T,0}, \sigma_{T,1}, \ldots, \sigma_{T,\psi_T - 1}\}$, add $(f_{T,\psi_T}, \sigma_{T,\psi_T})$ to L_T, returns σ_{T,ψ_T} to $\mathcal{A}$ and increment ψ_T by one.

At this point, $\mathcal{A}$ produces a challenge $(k, \sigma'_G, \sigma''_G, \sigma'''_G, \sigma'_T)$ such that (f'_G, σ'_G), $(f''_G, \sigma''_G), (f'''_G, \sigma'''_G) \in L_G$, and $(f'_T, \sigma'_T) \in L_T$ and abort. Observe that, $(f'_G, f''_G, f'''_G, f'_T)$ are polynomials of α, γ and s where α, γ will be sampled by $\mathcal{C}$ and s by $\mathcal{A}$. The generic group model ensures that $\mathcal{C}$ can verify $f'_T = f'_G$, $f'''_G = f''_G \cdot \alpha^{n-k}$ and $f''_G \cdot \gamma^{-1} \cdot {f'_T}^{-1} \in \mathsf{Span}(f_{G,1}, f_{G,2}, \ldots, f_{G,n})$.

Let $\mathbf{v} = (\alpha, \gamma, r)$ denote the vector consisting of variables over which the polynomials are defined. Now the simulator chooses at random $\alpha^*, \gamma^*, r^* \xleftarrow{\$} \mathbb{Z}_p$. Let $\mathbf{v}^* = (\alpha^*, \gamma^*, r^*)$. $\mathcal{C}$ assigns $\mathbf{v}^*$ to the variables $\mathbf{v}$. The simulation provided by $\mathcal{C}$ is perfect unless for some i, j any of the following holds.

1. $f_{G,i}(\mathbf{v}^*) - f_{G,j}(\mathbf{v}^*) = 0$ or some $i \neq j$ but $f_{G,i} \neq f_{G,j}$.
2. $f_{H,i}(\mathbf{v}^*) - f_{H,j}(\mathbf{v}^*) = 0$ or some $i \neq j$ but $f_{H,i} \neq f_{H,j}$.
3. $f_{T,i}(\mathbf{v}^*) - f_{T,j}(\mathbf{v}^*) = 0$ or some $i \neq j$ but $f_{T,i} \neq f_{T,j}$.

We use Bad to denote the event that at least one of the above holds, and we'll try to bound the probability of Bad. If Bad does not happen, then the simulation was perfect. Let assume $\mathcal{A}$ has generated his challenge for the set S, with $|S| > k$. So if Bad, does not happen $\mathcal{A}$ has no advantage in guessing $\mathit{\Gamma}_S$ over a random guess. Now in the polynomial f'''_G the highest possible degree of α is n if $|S| \leq k$. If $\mathcal{A}$ tries to simulate any group token where $|S| > k$ then in the polynomial f'''_G the highest degree of α is greater than n. Notice that α^{n+i} for some $i \geq 1$ is independent of $(1, \alpha, \ldots, \alpha^n)$. Also $\mathcal{A}$ does not have access to $g^{\alpha^{n+i}}$, which is outside the span of pk. Thus, it has one and only option: do it itself. Probability that it guesses f'''_G having no info about $g^{\alpha^{n+i}}$ is negligible.

Now, we would like to bound the probability of Bad. This is where we utilize the result on random assignment of polynomial due to Schwartz [Sch80]. Roughly speaking, the result states that for an n-variate polynomial $F(x_1, \ldots, x_n) \in \mathbb{Z}_p[X_1, \ldots, X_n]$ of degree d, a random assignment $x_1, \ldots, x_n \overset{\$}{\leftarrow} \mathbb{Z}_p$ make the polynomial F evaluate to zero with probability at most d/p. For fixed i, j, $(f_{G,i} - f_{G,j})$ is a polynomial of degree at most $n+1$, hence zero at random $\mathbf{v}^*$ with probability at most $(n+2)/p$. For fixed i, j, $f_{H,i} - f_{H,j}$ is a polynomial of degree at most n, hence zero at random $\mathbf{v}^*$ with probability at most n/p. For fixed i, j, $f_{T,i} - f_{T,j}$ is a polynomial of degree at most $n(n+1)$, hence zero at random $\mathbf{v}^*$ with probability at most $n(n+2)/p$. There are totally $\binom{\psi_G}{2}$, $\binom{\psi_H}{2}$, $\binom{\psi_T}{2}$ pairs of polynomials from L_G, L_H and L_T respectively. Note that, $\mathcal{A}$ is allowed to make at most m queries we have. Thus, $\psi_G + \psi_H + \psi_T \leq m + 2n + 3$.

There are totally $\binom{\psi_G}{2}$, $\binom{\psi_H}{2}$, $\binom{\psi_T}{2}$ pairs of polynomials from L_G, L_H and L_T respectively. Note that, $\mathcal{A}$ is allowed to make at most m queries we have. Thus, $\psi_G + \psi_H + \psi_T \leq m + 2n + 3$. Then,

$$\Pr[\mathsf{Bad}] \leq \binom{\psi_G}{2}(n+2)/p + \binom{\psi_H}{2} n/p + \binom{\psi_T}{2} n(n+2)/p \leq (m+2n+3)^2 \cdot \frac{n^2+4n+2}{2p}.$$

So $\Pr[\neg\mathsf{Bad}] = 1 - (m+2n+3)^2 \cdot \frac{n^2+4n+2}{2p}$ and $\Pr[\mathcal{A} \text{ wins}|\neg\mathsf{Bad}] = \frac{2}{2p-(m+2n+3)^2\cdot(n^2+4n+2)}$.

Now

$$\Pr[\mathcal{A} \text{ wins}] = \Pr[\mathcal{A} \text{ wins}|\neg\mathsf{Bad}]\Pr[\neg\mathsf{Bad}] + \Pr[\mathcal{A} \text{ wins}|\mathsf{Bad}]\Pr[\mathsf{Bad}].$$

So if Bad does not happen then $\mathcal{A}$ "knows" nothing about those possible values where any two $f_{b,i}(x) = f_{b,j}(x)$ happen for $1 \leq i < j \leq \psi_b$. Considering all this together probability that $\mathcal{A}$ wins is at most $\mathcal{O}(m^2/p)$. □

5.2.3 Message Indistinguishability from Dealer Let us consider three random encoding functions $\sigma_G : \mathbb{G} \to \{0,1\}^{m_G}$, $\sigma_H : \mathbb{H} \to \{0,1\}^{m_H}$ and $\sigma_T : \mathbb{G}_T \to \{0,1\}^{m_T}$ where w.l.o.g. $m_G \leq m_H \leq m_T$.

Theorem 3. *Let $\mathcal{A}$ be a* ppt *adversary against the message indistinguishability from dealer security game of $\mathsf{bed}_\mathcal{C}$ in the generic group model. Let n be any natural number and m be a bound on the total number of group elements $\mathcal{A}$ receives from queries it makes to the oracles computing the group actions in $\mathbb{G}$, $\mathbb{H}$, $\mathbb{G}_T$ and the bilinear map e. Then we have the advantage of $\mathcal{A}$ in the* message indistinguishability from dealer *security game of $\mathsf{bed}_\mathcal{C}$ is at most*

$$(m+2n+3)^2 \cdot \frac{n^2+4n+2}{4p}.$$

Proof. Let $\mathcal{C}$ denote an algorithm that simulates the generic bilinear group for $\mathcal{A}$. To answer oracle queries, $\mathcal{C}$ maintains three lists,

$$L_G = \{(f_{G,i}, \sigma_{G,i}) : i \in [0, \psi_G - 1]\}$$
$$L_H = \{(f_{H,i}, \sigma_{H,i}) : i \in [0, \psi_H - 1]\}$$
$$L_T = \{(f_{T,i}, \sigma_{T,i}) : i \in [0, \psi_T - 1]\}$$

such that at each step ψ of the game, the relation $\psi_G + \psi_H + \psi_T = \psi + 2n + 3$ holds. Here $f_{*,*}$ are multivariate polynomials over 5 variables α, γ, s, y_0, y_1 and $\sigma_{b,i}$ are strings from $\{0,1\}^{m_b}$, where $b \in \{G, H, T\}$. In the course of this proof we use $b \in \{G, H, T\}$ in the subscript to denote a representative of a general groups if not mentioned otherwise. At the beginning of the game i.e., $\psi = 0$, the lists are initialized by setting $\psi_G = (n+2)$, $\psi_H = n$ and $\psi_T = 1$. The polynomials $1, \alpha, \ldots, \alpha^n$ and γ are assigned to $f_{G,0}, f_{G,1}, \ldots, f_{G,n}, f_{G,n+1}$; $\alpha, \ldots, \alpha^n$, are assigned to $f_{H,1}, f_{H,2}, \ldots, f_{H,n}$; and γ is assigned to $f_{T,0}$.

These encodings for these polynomials are strings uniformly chosen from $\{0,1\}^{m_b}$ without repetition for polynomials $f_{b,*}$. We assume that $\mathcal{A}$ queries the oracles on strings previously obtained from $\mathcal{C}$ and naturally $\mathcal{C}$ can obtain the index of a given string $\sigma_{b,i}$ in the list L_b. The oracles are simulated as follows.

Group Actions in $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{G}_T$. We describe this for the group $\mathbb{G}$. We note that the group actions in $\mathbb{H}$ and $\mathbb{G}_T$ are simulated similarly. If $\mathcal{A}$ submits two strings $\sigma_{G,i}$ and $\sigma_{G,j}$ and a sign bit indicating addition or subtraction. $\mathcal{C}$ first finds $f_{G,i}$ and $f_{G,j}$ corresponding to $\sigma_{G,i}$ and $\sigma_{G,j}$ respectively in L_G and computes $f_{G,\psi_G} = f_{G,i} \pm f_{G,j}$. If there exists an index $k \in [0, \psi_G - 1]$, such that $f_{G,\psi_G} = f_{G,k}$, $\mathcal{C}$ sets $\sigma_{G,\psi_G} = \sigma_{G,k}$; otherwise $\mathcal{C}$ sets $\sigma_{G,\psi_G} \xleftarrow{\$} \{0,1\}^{m_G} \setminus \{\sigma_{G,0}, \sigma_{G,1}, \ldots, \sigma_{G,\psi_G - 1}\}$, add $(f_{G,\psi_G}, \sigma_{G,\psi_G})$ to L_G, returns σ_{G,ψ_G} to $\mathcal{A}$ and increment ψ_G by one.

Bilinear Map. If $\mathcal{A}$ submits two strings $\sigma_{G,i}$ and $\sigma_{H,j}$, $\mathcal{C}$ first finds $f_{G,i}$ in L_G corresponding to $\sigma_{G,i}$ and $f_{H,j}$ in L_H corresponding to $\sigma_{H,j}$ respectively and computes $f_{T,\psi_T} = f_{G,i} \pm f_{H,j}$. If there exists an index $k \in [0, \psi_T - 1]$, such that $f_{T,\psi_T} = f_{T,k}$, $\mathcal{C}$ sets $\sigma_{T,\psi_T} = \sigma_{T,k}$; otherwise $\mathcal{C}$ sets $\sigma_{T,\psi_T} \xleftarrow{\$} \{0,1\}^{m_T} \setminus \{\sigma_{T,0}, \sigma_{T,1}, \ldots, \sigma_{T,\psi_T - 1}\}$, add $(f_{T,\psi_T}, \sigma_{T,\psi_T})$ to L_T, returns σ_{T,ψ_T} to $\mathcal{A}$ and increment ψ_T by one.

At this point, $\mathcal{A}$ produces a challenge $(k, \sigma'_G, \sigma''_G, \sigma'''_G, \sigma'_T)$ such that (f'_G, σ'_G), $(f''_G, \sigma''_G), (f'''_G, \sigma'''_G) \in L_G$, and $(f'_T, \sigma'_T) \in L_T$. Observe that, $(f'_G, f''_G, f'''_G, f'_T)$ are polynomials of α, γ and s where α, γ will be sampled by $\mathcal{C}$ and s by $\mathcal{A}$. The generic group model ensures that $\mathcal{C}$ can verify $f'_T = f'_G$, $f'''_G = f''_G \cdot \alpha^{n-k}$ and $f''_G \cdot \gamma^{-1} \cdot {f'_T}^{-1} \in \mathsf{Span}(f_{G,1}, f_{G,2}, \ldots, f_{G,n})$. Finally, $\mathcal{C}$ sample $\beta \xleftarrow{\$} \{0,1\}$ and computes $\hat{f}'_G = f'_G \cdot y_0$, $\hat{f}''_G = f''_G \cdot y_0$ and $\hat{f}_T = f'_T \cdot y_\beta$. $\mathcal{C}$ then adds $(\hat{f}'_G, \hat{\sigma}'_G), (\hat{f}''_G, \hat{\sigma}''_G) \in L_G$, and $(\hat{f}_T, \hat{\sigma}_T) \in L_T$ following the above rules on group actions. At this point, $\mathcal{C}$ returns $(\hat{\sigma}'_G, \hat{\sigma}''_G, \hat{\sigma}_T)$ to $\mathcal{A}$ who returns β'.

Let $\mathbf{v} = (\alpha, \gamma, y_0, y_1)$ denote the vector consisting of variables over which the polynomials are defined. Now the simulator chooses at random $\alpha^*, \gamma^*, y_0^*, y_1^* \xleftarrow{\$} \mathbb{Z}_p$. Let $\mathbf{v}^* = (\alpha^*, \gamma^*, y_0^*, y_1^*)$. $\mathcal{C}$ assigns $\mathbf{v}^*$ to the variables $\mathbf{v}$. The simulation provided by $\mathcal{C}$ is perfect unless for some i, j any of the following holds.

1. $f_{G,i}(\mathbf{v}^*) - f_{G,j}(\mathbf{v}^*) = 0$ or some $i \neq j$ but $f_{G,i} \neq f_{G,j}$.
2. $f_{H,i}(\mathbf{v}^*) - f_{H,j}(\mathbf{v}^*) = 0$ or some $i \neq j$ but $f_{H,i} \neq f_{H,j}$.
3. $f_{T,i}(\mathbf{v}^*) - f_{T,j}(\mathbf{v}^*) = 0$ or some $i \neq j$ but $f_{T,i} \neq f_{T,j}$.

We use Bad to denote the event that at least one of the above holds and give the argument for the security proof in steps. First, we show that if Bad does not happen, the adversary $\mathcal{A}$ will have no advantage in winning the game over a random guess. Precisely, for a $\beta \xleftarrow{\$} \{0,1\}$, if $\mathcal{A}$ produces β' then $\Pr[\beta = \beta' : \neg\mathsf{Bad}] = 1/2$. To see this, observe that all variables except y_β and $y_{1-\beta}$ are independent of the bit β. Assume $y_\beta = r$ and recall $\mathcal{A}$ has access to all the lists (L_G, L_H, L_T) and gets $(\hat{\sigma}'_G, \hat{\sigma}''_G, \hat{\sigma}_T)$ as its challenge where $(\hat{f}'_G, \hat{\sigma}'_G), (\hat{f}''_G, \hat{\sigma}''_G) \in L_G$, and $(\hat{f}_T, \hat{\sigma}_T) \in L_T$. Observe that, $\hat{f}'_G, \hat{f}''_G, \hat{f}_T$ are respectively $y_0^* s$, $\gamma^* y_0^* s P_S(\alpha^*)$ and $y_\beta^* s$. Where $P_S(\alpha)$ is a polynomial of degree at most n. It is clear that $\hat{f}_T$ is a two-degree polynomial defined in the group $\mathbb{G}_T$. To compute such $\hat{f}_T$, we mention that $\mathcal{A}$ can not use the polynomial lists L_G and L_H and challenge polynomials $\hat{f}'_G$ and $\hat{f}''_G$. The only two-degree polynomials that can be constructed combining one polynomial from L_G and one from L_H are α^2 and $\alpha\gamma$. Thus, the best $\mathcal{A}$ can do is to output its guess β' at random and subsequently $\Pr[\beta = \beta' : \neg\mathsf{Bad}] = 1/2$.

Now, we would like to bound the probability of Bad. This is where we utilize the result on random assignment of polynomial due to Schwartz [Sch80]. Roughly speaking, the result states that for an n-variate polynomial $F(x_1, \ldots, x_n) \in \mathbb{Z}_p[X_1, \ldots, X_n]$ of degree d, a random assignment $x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p$ make the polynomial F evaluate to zero with probability at most d/p. For fixed i, j, $f_{G,i} - f_{G,j}$ is a polynomial of degree at most $n+1$, hence zero at random $\mathbf{v}^*$ with probability at most $(n+2)/p$. For fixed i, j, $f_{H,i} - f_{H,j}$ is a polynomial of degree at most n, hence zero at random $\mathbf{v}^*$ with probability at most n/p . For fixed i, j, $f_{T,i} - f_{T,j}$ is a polynomial of degree at most $n(n+1)$, hence zero at random $\mathbf{v}^*$ with probability at most $n(n+2)/p$. There are totally $\binom{\psi_G}{2}$, $\binom{\psi_H}{2}$, $\binom{\psi_T}{2}$ pairs of polynomials from L_G, L_H and L_T respectively. Note that, $\mathcal{A}$ is allowed to make at most m queries we have. Thus, $\psi_G + \psi_H + \psi_T \leq m + 2n + 3$. Then,

$$\Pr[\mathsf{Bad}] \leq \binom{\psi_G}{2}(n+1)/p + \binom{\psi_H}{2} n/p + \binom{\psi_T}{2} n(n+1)/p \leq (m + 2n + 3)^2 \cdot \frac{n^2 + 4n + 2}{2p}.$$

Now, a simple argument shows that,

$$\begin{aligned}\Pr\left[\beta=\beta'\right] &= \Pr\left[\beta=\beta'|\neg\mathsf{Bad}\right]\Pr\left[\neg\mathsf{Bad}\right]+\Pr\left[\beta=\beta'|\mathsf{Bad}\right]\Pr\left[\mathsf{Bad}\right]\\ &\leq \Pr\left[\beta=\beta'|\neg\mathsf{Bad}\right](1-\Pr\left[\mathsf{Bad}\right])+\Pr\left[\mathsf{Bad}\right]\\ &= \frac{1}{2}+\frac{1}{2}\Pr\left[\mathsf{Bad}\right]\end{aligned}$$

Also,

$$\Pr\left[\beta=\beta'\right]\geq\Pr\left[\beta=\beta'|\neg\mathsf{Bad}\right](1-\Pr\left[\mathsf{Bad}\right])=\frac{1}{2}-\frac{1}{2}\Pr\left[\mathsf{Bad}\right].$$

This two results were combined together to gives us

$$\mathsf{Adv}^{\mathsf{cpa}_{\mathcal{D}}}_{\mathcal{A},\mathsf{bed}_{\mathcal{C}}}=\left|\Pr\left[\beta=\beta'\right]-\frac{1}{2}\right|\leq\frac{\Pr\left[\mathsf{Bad}\right]}{2}\leq(m+2n+3)^2\cdot\frac{n^2+4n+2}{4p}.$$

□

5.2.4 Message Indistinguishability from Unprivileged Users

Theorem 4. *Let there exist a* ppt *adversary $\mathcal{A}$ who can break the selective message indistinguishability from unprivileged users of* $\mathsf{bed}_{\mathcal{C}}$ *with the non-negligible advantage, then it is possible to construct an efficient adversary $\mathcal{B}$ that has a non-negligible advantage in breaking* AS_1 *and* AS_2 *in* $\mathcal{CG}=(p,q,g_p,g_q,h_p,h_q,\mathbb{G},\mathbb{H},\mathbb{G}_T,e)\leftarrow\mathsf{CBGen}(1^\lambda)$.

Proof. This proof is done via a sequence of games. Precisely, we follow [GLR18] to apply Deja Q framework [CM14,Wee16] to prove the security of BED in the standard model. Let us use S^* to denote the target user set and $x_1,\ldots,x_Q$ to denote the corrupted users.

Game_0. This is same as the real game.

Game_1. The following natural assumptions are made on the game.

- For all $z\in S^*\cup\{x_1\}\cup\ldots\cup\{x_Q\}$, $(\alpha+z)$ is not divisible by p. Otherwise, $\mathcal{B}$ can easily solve the subgroup decision problem AS_1 by computing $\mathsf{gcd}((\alpha+z),N)$.
- For all $i,j\in[q]$ and $i\neq j$, if $x_i\neq x_j \mod N$ then $x_i\neq x_j \mod q$. Otherwise, $\mathcal{B}$ can easily solve the subgroup decision problem AS_2 by computing $\mathsf{gcd}((x_i-x_j),N)$.

Therefore, $|\Pr[X_1]-\Pr[X_0]|\leq\mathsf{Adv}^{\mathsf{AS}_1}_{\mathcal{B}}(\lambda)+\mathsf{Adv}^{\mathsf{AS}_2}_{\mathcal{B}}(\lambda)$.

Game_2. We perform a conceptual change to Game_1 here. Given the challenge user set S^* of size k, pick $\alpha,\widetilde{\gamma},h_p\xleftarrow{\$}\mathbb{Z}_N^2\times\mathbb{H}_p$. Define polynomial $F_{S^*}(z)=\prod\limits_{y\in S^*}(z+y)$ and set $\gamma=\widetilde{\gamma}\cdot F_{S^*}(\alpha) \mod N$. In pk, this affects only g_p^γ. The rest of the public parameters in pk are defined exactly the same as in Game_1. The secret keys corresponding to x_i is $\mathsf{sk}_{x_i}=h_p^{\frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha)}{(\alpha+x_i)}}$ for $i\in[Q]$. The challenge ciphertext is CT_{S^*} where $\mathsf{CT}_{S^*}=(\mathsf{ct}_1,\mathsf{ct}_2,\mathsf{ct}_0)$ such that

$$\mathsf{ct}_1=g_p^{t\widetilde{\gamma}F_{S^*}(\alpha)},\mathsf{ct}_2=g_p^{tF_{S^*}(\alpha)}=\mathsf{ct}_1^{1/\widetilde{\gamma}},\mathsf{ct}_0=M\oplus\mathsf{H}(e(\mathsf{ct}_1,h_p)).$$

Note that, the replacement $\gamma=\widetilde{\gamma}\cdot F_{S^*}(\alpha) \mod N$ doesn't change the ciphertext distribution as $\widetilde{\gamma}$ is uniformly random and $F_{S^*}(\alpha)\neq 0 \mod p$. Therefore, $\Pr[X_2]=\Pr[X_1]$.

Game_3. Another conceptual change to Game_2 is performed here. Choose $\mathsf{ct}_1\xleftarrow{\$}\mathbb{G}_p$. The rest of the ciphertext is defined the same as in Game_2. As both ct_0 and ct_2 are functions of ct_1, namely $\mathsf{ct}_0=M\oplus\mathsf{H}(e(\mathsf{ct}_1,h_p))$ and $\mathsf{ct}_2=\mathsf{ct}_1^{1/\widetilde{\gamma}}$, such a replacement doesn't change the distribution of the challenge ciphertext or the challenge encapsulation key. Therefore, $\Pr[X_3]=\Pr[X_2]$.

Game_4**.** Here the subgroup decision assumption AS_1 is used to choose ct_1 from the group $\mathbb{G}$ uniformly at random. Other ciphertext components and secret keys are generated similar to Game_3. Therefore, $|\Pr[X_4] - \Pr[X_3]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{AS}_1}(\lambda)$. We provide an informal argument here. Given the problem instance AS_1, $\mathcal{B}$ chooses $\alpha, \widetilde{\gamma} \xleftarrow{\$} \mathbb{Z}_N$. This allows $\mathcal{B}$ to compute all of pk similar to Game_3. As $\mathcal{B}$ holds both α and $\widetilde{\gamma}$, it can answer any key extraction query. In the challenge phase, it uses the target T of AS_1 problem instance to simulate CT_{S^*}. If T was from $\mathbb{G}_p$, ct_1 is normal whereas if T was from $\mathbb{G}$, then ct_1 is semi-functional. Since ct_1 determines the challenge ciphertext completely, the distribution from which T was chosen determines if the challenge ciphertext is normal or semi-functional.

Game_5**.** Here we change the distribution of a few public parameters $u_1 = h_p^{\alpha}, \ldots, u_n = h_p^{\alpha^n}$ and the secret keys $\mathsf{sk}_{x_1}, \ldots, \mathsf{sk}_{x_Q}$. We modify $u_0 = h_p$ as well. This is done via intermediate games $\{\mathsf{Game}_{5,k,0}, \mathsf{Game}_{5,k,1}\}_{k \in [n+Q+1]}$. Note that, we define $\mathsf{Game}_{5,k,0} = \mathsf{Game}_4$ and $\mathsf{Game}_{5,n+Q+1,1} = \mathsf{Game}_5$.

- In $\mathsf{Game}_{5,k,0}$ we make following changes to u_i for $i \in [0, n]$.

$$h_p^{\alpha^i} \cdot h_q^{\sum_{j \in [k-1]} r_j \alpha_j^i} \to h_p^{\alpha^i} \cdot \boxed{h_q^{r\alpha^i}} \cdot h_q^{\sum_{j \in [k-1]} r_j \alpha_j^i} \tag{4}$$

Then we also make changes to sk_{x_i} for $i \in [Q]$

$$h_p^{\frac{\widetilde{\gamma} \cdot F_{S^*}(\alpha)}{(\alpha + x_i)}} \cdot h_q^{\sum_{j \in [k-1]} \frac{r_j \cdot \widetilde{\gamma} \cdot F_{S^*}(\alpha_j)}{(\alpha + x_i)}} \to h_p^{\frac{\widetilde{\gamma} \cdot F_{S^*}(\alpha)}{(\alpha + x_i)}} \cdot \boxed{h_q^{\frac{r \cdot \widetilde{\gamma} \cdot F_{S^*}(\alpha)}{(\alpha + x_i)}}} \cdot h_q^{\sum_{j \in [k-1]} \frac{r_j \cdot \widetilde{\gamma} \cdot F_{S^*}(\alpha_j)}{(\alpha + x_i)}} \tag{5}$$

The boxed parts denote the modification $\mathsf{Game}_{5,k,0}$ introduces. We now show that this modification is invisible to the adversary $\mathcal{A}$ via the following lemma.

Lemma 1. *There exists a* ppt *adversary such that* $|\Pr[X_{\mathsf{Game}_{5,k,0}}] - \Pr[X_{\mathsf{Game}_{5,k-1,1}}]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{AS}_2}(\lambda)$.

Proof. The solver $\mathcal{B}$ is given the problem instance $D = (g_p, h_p, g_p^a g_q^b, h_p^z)$ and the target T.

Setup. The adversary $\mathcal{A}$ sends the challenger target set S^*. $\mathcal{B}$ chooses $\alpha, \widetilde{\gamma} \xleftarrow{\$} \mathbb{Z}_N^2$ to generate the public parameters $g_p^{\alpha}, \ldots, g_p^{\alpha^n}, g_p^{\gamma}$ efficiently where $\gamma = \widetilde{\gamma} \cdot F_{S^*}(\alpha) \mod N$. It then chooses $r_1, \alpha_1, \ldots, r_{k-1}, \alpha_{k-1} \xleftarrow{\$} \mathbb{Z}_N$. The public parameters $u_1, \ldots, u_n$ are generated as follows along with $u_0 = h_p$ which is used to compute $e(g_p, h_p)^{\gamma} = e(g_p, u_0)^{\gamma}$. For

$$u_i = T^{\alpha^i} h_q^{\sum_{j \in [k-1]} r_j \alpha_j^i}.$$

$\mathcal{B}$ then outputs public parameter

$$\mathsf{pk} = (g_p, g_p^{\alpha}, \ldots, g_p^{\alpha^n}, g_p^{\gamma}, u_1, \ldots, u_n e(g_p, u_0)^{\gamma}, \mathsf{H}),$$

where H is randomly chosen universal hash function.

Phase-I Queries. On a secret key query on x_i, $\mathcal{B}$ sets

$$\mathsf{sk}_{x_i} = T^{\frac{\widetilde{\gamma} \cdot F_{S^*}(\alpha)}{\alpha + x_i}} \cdot h_q^{\sum_{j \in [k-1]} \frac{r_j \cdot \widetilde{\gamma} \cdot F_{S^*}(\alpha_j)}{(\alpha_j + x_i)}}.$$

Challenge. $\mathcal{B}$ here computes $\mathsf{CT}_{S^*} = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_0)$ where $\mathsf{ct}_1 = g_p^a g_q^b$, $\mathsf{ct}_2 = \mathsf{ct}_1^{1/\widetilde{\gamma}}$ and $\mathsf{ct}_0^{(0)} = M \oplus \mathsf{H}(e(\mathsf{ct}_1, u_0))$. $\mathcal{B}$ chooses $\mathsf{ct}_0^{(1)} \xleftarrow{\$} \mathcal{K}$ and outputs $(\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_0^{(\beta)})$ for $\beta \xleftarrow{\$} \{0, 1\}$.

Phase-II Queries. Same as Phase-I queries.

Guess. $\mathcal{B}$ outputs 1 if Adv's guess β' is same as $\mathcal{B}$'s choice β.

If $T \in \mathbb{H}_p$, then the game distribution is same as $\mathsf{Game}_{5,k-1,1}$. On the other hand, if $T \in \mathbb{H}$, then the game distribution is same as $\mathsf{Game}_{5,k,0}$. □

– In $\mathsf{Game}_{5,k,1}$ we make following changes to to u_i for $i \in [0, n]$.

$$h_p^{\alpha^i} \cdot \boxed{h_q^{r\alpha^i}} \cdot h_q^{\sum_{j\in[k-1]} r_j\alpha_j^i} \to h_p^{\alpha^i} \cdot h_q^{\sum_{j\in[k]} r_j\alpha_j^i} \tag{6}$$

Then we also make changes to sk_{x_i} for $i \in [Q]$

$$h_p^{\frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha)}{(\alpha+x_i)}} \cdot \boxed{h_q^{\frac{r\cdot\widetilde{\gamma}\cdot F_{S^*}(\alpha)}{(\alpha+x_i)}}} \cdot h_q^{\sum_{j\in[k-1]} \frac{r_j\cdot\widetilde{\gamma}\cdot F_{S^*}(\alpha_j)}{(\alpha+x_i)}} \to h_p^{\frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha)}{(\alpha+x_i)}} \cdot h_q^{\sum_{j\in[k]} \frac{r_j\cdot\widetilde{\gamma}\cdot F_{S^*}(\alpha_j)}{(\alpha+x_i)}} \tag{7}$$

The boxed parts above denote the components where $\mathsf{Game}_{5,k,1}$ makes the modification. Precisely, we replace α and r in the exponent of h_q with $\alpha_k \mod q$ and r_k respectively. This does not affect the view of the adversary as none of the public parameters or the ciphertext reveals $\alpha \mod q$. Indeed due to the replacement in Game_3, the ciphertext does reveal $\alpha \mod p$ which by the Chinese Remainder Theorem (CRT) is independent of $\alpha \mod q$. We thus replace α and r in the exponent of h_q with α_k mod q and r_k respectively without the adversary noticing it and this is a conceptual change.

Game_6. Here we replace $\mathbb{H}_q$ components of $\{u_i\}_{i\in[0,n]}$ and the secret keys $\{\mathsf{sk}_{x_i}\}_{i\in[Q]}$ modified above. Namely, we now define them as following:

$$u_i = h_p^{\alpha^i} \cdot h_q^{t_i}, \forall i \in [0, n] \qquad\qquad \mathsf{sk}_{x_i} = h_p^{\frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha)}{(\alpha+x_i)}} \cdot h_q^{t_{n+i}}, \forall i \in [Q]$$

We then argue that Game_5 and Game_6 are statistically close. This can be seen from the $\mathbb{H}_q$ components of $\{u_i\}_{i\in[0,n]}$ and $\{\mathsf{sk}_{x_i}\}_{i\in[Q]}$. Precisely, the exponents of the said elements in the $\mathbb{H}_q$ group can be represented by the following linear system of equations:

$$\begin{pmatrix} t_0 \\ t_1 \\ \vdots \\ t_n \\ t_{n+1} \\ \vdots \\ z_{n+Q} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{n+Q+1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^n & \alpha_2^n & \cdots & \alpha_{n+Q+1}^n \\ \frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha_1)}{\alpha_1+x_1} & \frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha_2)}{\alpha_2+x_1} & \cdots & \frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha_{n+Q+1})}{\alpha_{n+Q+1}+x_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha_1)}{\alpha_1+x_n} & \frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha_2)}{\alpha_2+x_n} & \cdots & \frac{\widetilde{\gamma}\cdot F_{S^*}(\alpha_{n+Q+1})}{\alpha_{n+Q+1}+x_n} \end{pmatrix}}_{\mathbf{A}} \cdot \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{n+Q+1} \end{pmatrix}. \tag{8}$$

Since, $x_i \notin S^*$ for all $i \in [Q]$, following [GLR18] the above matrix $\mathbf{A}$ can be transformed into $\mathbf{B}$ defined below such that $\mathsf{det}(\mathbf{A}) = \widetilde{\gamma}^q \cdot \mathsf{det}(\mathbf{B})$.

$$\mathbf{B} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{n+Q+1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^n & \alpha_2^n & \cdots & \alpha_{n+Q+1}^n \\ \frac{1}{\alpha_1+x_1} & \frac{1}{\alpha_2+x_1} & \cdots & \frac{1}{\alpha_{n+Q+1}+x_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_1+x_n} & \frac{1}{\alpha_2+x_n} & \cdots & \frac{1}{\alpha_{n+Q+1}+x_n} \end{pmatrix}. \tag{9}$$

Note that, $\mathsf{det}(\mathbf{B}) = \delta \cdot \frac{\prod_{1\le\ell<j\le Q}(x_\ell - x_j)\prod_{1\le i<k\le(n+Q+1)}(\alpha_i-\alpha_k)}{\prod_{k=1}^{(n+Q+1)}\prod_{\ell=1}^{Q}(\alpha_k+x_t)} \neq 0$ due to [GLR18,CM19]. Naturally $\mathsf{det}(\mathbf{B}) = 0$ if $\alpha_i = \alpha_k$ for distinct $i, k \in [n + Q + 1]$. Thus, $\Pr[\mathsf{det}(\mathbf{B}) = 0] \le (n + Q + 1)^2/q$. Therefore, $|\Pr[X_6] - \Pr[X_5]| \le (n + Q + 1)^2/q$.

Game_7. Now we replace $\mathsf{ct}_0 = M \oplus \mathsf{H}(e(\mathsf{ct}_1, u_0))$ by a uniform random choice from $\mathcal{K}$. The reason behind this is u_0 now is $h_p \cdot h_q^{t_0}$. As we saw in the last game, t_0 is a uniformly random quantity independent of all $t_1, \ldots, t_{(n+Q)}$. Thus $e(\mathsf{ct}_1, u_0) = e(\mathsf{ct}_1, h_p) \cdot e(\mathsf{ct}_1, h_q^{t_0})$ has $\log q$ bits of min-entropy due to $t_0 \mod q$. Due to left-over hash lemma [HILL99], H is a strong extractor [AB09] and therefore, $\mathsf{ct}_0 = M \oplus \mathsf{H}(e(\mathsf{ct}_1, u_0))$ is at most $2^{-\lambda}$ distance from the uniform distribution on $\mathcal{K}$ provided $\mathbb{G}_q$ component in ct_1 is not 1. The probability that the $\mathbb{G}_q$ component of ct_1 is 1 is $1/q$. Therefore, $|\Pr[X_7] - \Pr[X_6]| \leq 1/q + 2^{-\lambda}$. The encapsulation key $\mathsf{H}(e(\mathsf{ct}_1, u_0))$ now is a randomly distributed element from $\mathcal{K}$ and it hides β completely i.e. $\Pr[X_7] = 1/2$.

□

6 Adaptively Secure BED ($\mathsf{bed}_{\mathsf{anon}}$)

The previous construction achieved only selective message indistinguishability from unprivileged users. To improve upon the result, we start from a rather obvious observation that anonymous broadcast encryption hides the privileged set completely. Therefore, if one constructs a BED from anon-BE, the group privacy holds for free. Observe that, the message hiding from unprivileged users is essentially the ind-cpa security of the anon-BE. Based on these observations, we give our second BED construction ($\mathsf{bed}_{\mathsf{anon}}$) from the efficient anon-BE protocol of Li and Gong [LG18] that achieved adaptive anonymity in the standard model. In particular, we modify their construction for our purpose and argue the necessary security properties. We here mention that similar to [LG18], we also assume that there is an efficient symmetric-key encryption SE which is key-binding [Fis99].

6.1 Construction

It is defined by the following ppt algorithms.

- $\mathsf{Setup}(1^\lambda, n)$:
  1. Run $\mathcal{G} = (p, g, \mathbb{G}) \xleftarrow{\$} \mathsf{Ggen}(1^\lambda)$.
  2. Choose $\alpha, u_i, v_i \leftarrow \mathbb{Z}_p$ for $i \in [n]$.
  3. Set $\mathsf{msk} = \Big(\{(u_i, v_i)\}_{i \in [n]}\Big)$.
  4. Publish $\mathsf{pk} = (g, g^\alpha, g^{u_1+\alpha v_1}, g^{u_2+\alpha v_2}, \ldots, g^{u_n+\alpha v_n})$.
- $\mathsf{KeyGen}(\mathsf{msk}, i)$: Output $\mathsf{sk}_i = (u_i, v_i)$.
- $\mathsf{GroupGen}(\mathsf{pk}, k, S)$:
  1. Suppose $S = \{i_1, \ldots, i_{k'}\}$ where $k' \leq k \leq n$.
  2. Define a random permutation τ on S such that $\tau(i_j) = i'_j \in S$ for $j \in [k']$.
  3. Choose $s \xleftarrow{\$} \mathbb{Z}_p$ and $\kappa \xleftarrow{\$} \mathcal{K}$.
  4. Set $\mathit{\Gamma}_S = \big(h_0, h_1, \{\omega_i\}_{i \in [k']}, \kappa\big)$ for
  $$h_0 = g^s; \qquad h_1 = g^{s\alpha}; \qquad \omega_i = g^{s(u_i+\alpha v_i)} \cdot \kappa, \forall i \in [k'].$$
- $\mathsf{Verify}(\mathsf{pk}, \mathit{\Gamma}_S, k)$: If $|\mathit{\Gamma}_S| \leq k - 2$, output 1. Otherwise, output 0.
- $\mathsf{Enc}(\mathsf{pk}, \mathit{\Gamma}_S, M)$:
  1. Parse $\mathit{\Gamma}_S = (h_0, h_1, \omega_1, \ldots, \omega_{k'}, \kappa)$.
  2. Sample $r \xleftarrow{\$} \mathbb{Z}_p$. Compute $\mathsf{ct} = \mathsf{SE.E}(\kappa^r, M)$, for $M \in \mathcal{M}$.
  3. Output $\mathsf{ct}_S = (\mathsf{ct}, H_0, H_1, c_1 \ldots, c_{k'}) = (\mathsf{SE.E}(\kappa^r, M), h_0^r, h_1^r, \omega_1^r, \ldots, \omega_{k'}^r)$. Note that ω_i^r also changes κ to κ^r.
- $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct}_S)$:
  1. Parse $\mathsf{ct}_S = (\mathsf{ct}, H_0, H_1, c_1 \ldots, c_{k'})$.
  2. For $j \in [|S|]$, compute $\frac{c_j}{H_0^{u_i} \cdot H_1^{v_i}}$ to get back κ_j.
  3. Let $M' = \mathsf{SE.D}(\kappa_j, \mathsf{ct})$. If $M' \neq \perp$, output M'.

Correctness. The correctness of the construction follows due to the key-binding property of the symmetric-key encryption SE. Precisely, if $\kappa_j \neq \kappa$, the key-binding property ensures that $\mathsf{SE.D}(\kappa_j, \mathsf{ct}) = \perp$.

6.2 Security

We prove the above construction achieves security in all four security models.

6.2.1 Group Privacy

Theorem 5. *Let there exists a* ppt *adversary* $\mathcal{A}$ *breaking the group privacy of* $\mathsf{bed_{anon}}$ *with a non-negligible advantage then there exists a* ppt *adversary* $\mathcal{B}$ *which has a non-negligible advantage in solving DDH problem in* $\mathbb{G}$ *where* $\mathcal{G} = (p, g, \mathbb{G}) \leftarrow \mathsf{PGen}(1^\lambda)$ *such that* $\mathbb{G} = \langle g \rangle$.

Proof. This proof is done via a sequence of games starting with the real construction Game_0. We assume X_i denotes the event the adversary winning Game_i.

Game_0**:** The Game_0 is the real construction. Here, the challenger samples $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and $u_i, v_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [n]$. It sets $\mathsf{pk} = (g, g^\alpha, g^{u_1+\alpha v_1}, g^{u_2+\alpha v_2}, \dots, g^{u_n+\alpha v_n})$. When the adversary gives distinct sets $S_0, S_1 \subset [n]$ of same cardinality k', the game samples $\beta \xleftarrow{\$} \{0,1\}$ and responds with the group token $\mathit{\Gamma}_{S_\beta} = \big(h_0, h_1, \{\omega_i\}_{i\in[k']}, \kappa\big)$ where

$$h_0 = g^s; \qquad h_1 = g^{s\alpha}; \qquad \omega_i = g^{s(u_i+\alpha v_i)} \cdot \kappa.$$

The adversary finally returns $\beta' \in \{0,1\}$ as its guess.

Game_1**:** This game is nothing but a conceptual change of Game_0. We modify the way ciphertext is produced here. Precisely, we use h_0 and h_1 to construct $\{\omega_i\}_{i\in[k']}$. When the adversary gives distinct sets $S_0, S_1 \subset [n]$ of same cardinality k', the game samples $\beta \xleftarrow{\$} \{0,1\}$ and responds with the group token $\mathit{\Gamma}_{S_\beta} = \big(h_0, h_1, \{\omega_i\}_{i\in[k']}, \kappa\big)$ where

$$h_0 = g^s; \qquad h_1 = g^{s\alpha}; \qquad \omega_i = h_0^{u_i} h_1^{v_i} \cdot \kappa.$$

Note that, $\Pr[X_1] = \Pr[X_0]$.

Game_2**:** In this game, we change the way h_1 is sampled. Precisely, we sample h_1 uniformly at random. When the adversary gives distinct sets $S_0, S_1 \subset [n]$ of same cardinality k', the game samples $\beta \xleftarrow{\$} \{0,1\}$ and responds with the group token $\mathit{\Gamma}_S = \big(h_0, h_1, \{\omega_i\}_{i\in[k']}, \kappa\big)$ where

$$h_0 = g^s; \qquad h_1 \xleftarrow{\$} \mathbb{G}; \qquad \omega_i = h_0^{u_i} h_1^{v_i} \cdot \kappa.$$

Note that, any adversary that can distinguish between Game_1 and Game_2, can be used to solve ddh problem.

Lemma 2. *There exists a* ppt *adversary such that* $|\Pr[X_2] - \Pr[X_1]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DDH}}(\lambda)$.

Proof. Given the DDH problem instance (g, g^α, g^s, T), we define $\mathcal{B}$ to choose $\alpha, u_1, v_1, \dots, u_n, v_n \xleftarrow{\$} \mathbb{Z}_p$. $\mathcal{B}$ then computes $\mathsf{pk} = (g, g^\alpha, g^{u_1}(g^\alpha)^{v_1}, \dots, g^{u_n}(g^\alpha)^{v_n})$.

When the adversary gives distinct sets $S_0, S_1 \subset [n]$ of same cardinality k', $\mathcal{B}$ samples $\beta \xleftarrow{\$} \{0,1\}$ and responds with the group token $\mathit{\Gamma}_{S_\beta} = \big(h_0, h_1, \{\omega_i\}_{i\in[k']}, \kappa\big)$ where

$$h_0 = g^s; \qquad h_1 = g^c; \qquad \omega_i = h_0^{u_i} h_1^{v_i} \cdot \kappa.$$

If $T = g^{\alpha s}$, $\mathcal{B}$ simulates Game_1 whereas if $T = g^c$ for $c \xleftarrow{\$} \mathbb{Z}_p$, then $\mathcal{B}$ simulates Game_2. □

Game_3**:** In this game, we change the way pk and ciphertext is computed. The challenger here samples $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and $u_i, v_i, \mu_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [n]$. It defines $u'_i = u_i - \alpha\mu_i$ and $v'_i = v_i + \mu_i$; and outputs $\mathsf{pk} = (g, g^\alpha, g^{u'_1}(g^\alpha)^{v'_1}, \ldots, g^{u'_n}(g^\alpha)^{v'_n})$. Observe that pk is exactly same as in Game_2. Precisely, $\{\mu_i\}_{i\in[n]}$ are not leaked from pk.

When the adversary gives distinct sets $S_0, S_1 \subset [n]$ of same cardinality k', the game samples $\beta \xleftarrow{\$} \{0,1\}$ and responds with the group token $\mathit{\Gamma}_{S_\beta} = \big(h_0, h_1, \{\omega_i\}_{i\in[k']}, \kappa\big)$ where

$$h_0 = g^s; \qquad h_1 = g^{(\alpha s+\delta)}; \qquad \omega_i = h_0^{u'_i} h_1^{v'_i} \cdot \kappa.$$

This is simply a conceptual change of the way pk and ciphertext are computed. Thus $\Pr[X_3]-\Pr[X_2] = 0$. Now observe that, $\omega_i = h_0^{u'_i} h_1^{v'_i} \cdot \kappa = h_0^{u_i} h_1^{v_i} g^{-\alpha s\mu_i} g^{\delta\mu_i+\alpha s\mu_i}\kappa = h_0^{u_i} h_1^{v_i} g^{\delta\mu_i}\kappa$. Since, μ_i are uniformly random element, $g^{\delta\mu_i}$ are purely random element that hides u_i and v_i for all $i \in S_\beta$. Thus, $\Pr[X_3] = 1/2$.

□

6.2.2 Maximum Users Accountability

Observe that, $\mathit{\Gamma}_S$ leaks the cardinality of S. Therefore, maximum users accountability follows trivially.

6.2.3 Message Indistinguishability from Dealer

Let us consider a random encodings function $\sigma : \mathbb{G} \to \{0,1\}^{m_G}$.

Theorem 6. *Let $\mathcal{A}$ be a* ppt *adversary against the message indistinguishability from dealer security of* $\mathsf{bed_{anon}}$ *in the generic group model. Let n be any natural number and m be a bound on the total number of group elements $\mathcal{A}$ receives from queries it makes to the oracles computing the group actions in $\mathbb{G}$. Then we have that the advantage of $\mathcal{A}$ in the* message indistinguishability from dealer *security game of* $\mathsf{bed_{anon}}$ *is at most* $(m+n+2)^2 \cdot \frac{3}{4p}$.

Proof. Let $\mathcal{C}$ be the algorithm that simulates the generic bilinear group for $\mathcal{A}$. To answer oracle queries, $\mathcal{C}$ maintains a list

$$L = \{(f_i, \sigma_i) : i \in [0, \psi_G - 1]\}$$

such that at each step ψ of the game, the relation $\psi_G = \psi + n + 2$ holds. Here f_i are multivariate polynomials over $(2n+3)$ variables α, $k_1^{(i)}$, $k_2^{(i)}$, s, y_0, y_1, for $i \in [n]$ and σ_i are strings from $\{0,1\}^{m_G}$. At the beginning of the game i.e., $\psi = 0$, the lists are initialized by setting $\psi_G = n+2$. The polynomials 1, α, $\{(k_1^{(i)} + \alpha k_2^{(i)})_{i\in[n]}\}$ are assigned to f_0, f_1, $\ldots$, f_{n+1}. These encodings for these polynomials are strings uniformly chosen from $\{0,1\}^{m_G}$ without repetition for polynomials f_i where $i \in \{0, \ldots, n+1\}$. We assume that $\mathcal{A}$ queries the oracles on strings previously obtained from $\mathcal{C}$ and naturally $\mathcal{C}$ can obtain the index of a given string σ_i in the list L. The oracles are simulated as follows.

Group Actions in $\mathbb{G}$. We describe how the group action for $\mathbb{G}$ is simulated. If $\mathcal{A}$ submits two strings σ_i and σ_j and a sign bit indicating addition or subtraction. $\mathcal{C}$ first finds f_i and f_j corresponding to σ_i and σ_j respectively in L and computes $f_{\psi_G} = f_i \pm f_j$. If there exists an index $k \in [0, \psi_G - 1]$, such that $f_{\psi_G} = f_k$, $\mathcal{C}$ sets $\sigma_{\psi_G} = \sigma_k$; otherwise $\mathcal{C}$ sets $\sigma_{\psi_G} \xleftarrow{\$} \{0,1\}^{m_G} \setminus \{\sigma_0, \sigma_1, \ldots, \sigma_{\psi_G - 1}\}$ save (f_{ψ_G}, σ_G) to L, returns σ_{ψ_G} to $\mathcal{A}$ and increment ψ_G by one.

At this point, $\mathcal{A}$ produces a challenge $(k, \sigma'_0, \sigma'_1, \sigma'_1, \ldots, \sigma'_{k'+1})$ such that (f'_0, σ'_0), $(f'_1, \sigma'_1), (f'_2, \sigma'_2), \ldots, (f'_{k'+1}, \sigma'_{k'+1}) \in L$. Observe that, $(f'_0, f'_1, f'_2, \ldots, f'_{k'+1})$ are polynomials of α, k_1, k_2, and s where α, k_1, k_2 will be sampled by $\mathcal{C}$ and s by $\mathcal{A}$ The generic group model ensures that $\mathcal{C}$ can verify $f'_0 = f'_1 \cdot \alpha^{-1}$, and $f'_i \in \mathsf{Span}(f_0, f_1, f_2, \ldots, f_{n+1})$, $\forall i \in [2, k'+1]$. Finally, $\mathcal{C}$ samples $\beta \xleftarrow{\$} \{0,1\}$, computes $\hat{f}'_0 = f'_0 \cdot y_0$, $\hat{f}'_1 = f'_1 \cdot y_0$, and $\hat{f}'_i = f'_i \cdot y_\beta$, then adds $(\hat{f}'_0, \hat{\sigma}'_0), (\hat{f}'_1, \hat{\sigma}'_1), \ldots, (\hat{f}'_{k'+1}, \hat{\sigma}'_{k'+1}) \in L$ following the above rules on group actions. At this point, $\mathcal{C}$ returns $(\hat{\sigma}'_0, \hat{\sigma}'_1, \hat{\sigma}'_2, \ldots, \hat{\sigma}'_{k'+1})$ to $\mathcal{A}$ who returns β' and abort.

Let $\mathbf{v} = \left(\alpha, k_1^{(i)}, k_2^{(i)}, y_0, y_1\right)$ denote the vector consisting of variables over which the polynomials are defined. Now the simulator chooses at random $\alpha^*, k_1^{(i)^*}, k_2^{(i)^*}, y_0^*, y_1^* \xleftarrow{\$} \mathbb{Z}_p$, for all $i \in [n]$. Let $\mathbf{v} = \left(\alpha^*, k_1^{(i)^*}, k_2^{(i)^*}, y_0^*, y_1^*\right)$. $\mathcal{C}$ assigns $\mathbf{v}^*$ to the variables $\mathbf{v}$. The simulation provided by $\mathcal{C}$ is perfect unless for some $i, j \le \psi_b$ the following holds.

$$f_i(\mathbf{v}^*) - f_j(\mathbf{v}^*) = 0 \text{ for some } i \neq j \text{ but } f_i \neq f_j.$$

We use Bad to denote the above hold and we'll try to bound the probability of Bad. If Bad does not happen, then the simulation was perfect. So if Bad, does not happen $\mathcal{A}$ has no advantage in guessing β' over a random guess. Precisely, $\Pr\left[\beta = \beta' | \neg\mathsf{Bad}\right] = 1/2$. To see this, observe that all variables except y_β and $y_{1-\beta}$ are independent of the bit β. Assume $y_\beta = r$ and recall $\mathcal{A}$ has access to L and gets $(\hat{\sigma}'_0, \hat{\sigma}'_1, \hat{\sigma}'_2, \ldots, \hat{\sigma}'_{k'+1})$ as its challenge. where $(\hat{f}'_0, \hat{\sigma}'_0), (\hat{f}'_1, \hat{\sigma}'_1), \ldots, (\hat{f}'_{k'+1}, \hat{\sigma}'_{k'+1}) \in L$. Observe that, $\hat{f}'_0, \hat{f}'_1, \hat{f}'_2, \ldots, \hat{f}'_{k'+1}$ are respectively sy_0^*, $s\alpha^* y_0^*$, $sy_\beta^*(k_1^{(1)} + \alpha k_2^{(1)}), \ldots, sy_\beta^*(k_1^{(k')} + \alpha k_2^{(k')})$. It is clear that $\hat{f}'_i$ for $i \in [2, k'+1]$ is a degree four polynomial but both $\hat{f}'_0$ and $\hat{f}'_1$ are polynomials of degree two and three. To compute such $\hat{f}'_i$, we mention that $\mathcal{A}$ can use the polynomial lists L and challenge polynomials. Indeed, taking linear composition of $\hat{f}'_0$ and $\hat{f}'_1$ with L results in polynomials of three. All the degree four polynomials that can only be computed does involve only y_β^*. Thus, the best $\mathcal{A}$ can do is to output its guess β' at random and subsequently $\Pr[\beta = \beta' : \neg\mathsf{Bad}] = 1/2$.

Now, we show that the probability that Bad happens is negligible. This is where we utilize the result on random assignment of polynomial due to Schwartz [Sch80]. Roughly speaking, the result states that for an n-variate polynomial $F(x_1, \ldots, x_n) \in \mathbb{Z}_p[X_1, \ldots, X_n]$ of degree d, a random assignment $x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p$ make the polynomial F evaluate to zero with probability at most d/p. For fixed i, j, $f_i - f_j$ is a polynomial of degree at most 3, hence zero at random $\mathbf{v}^*$ with probability at most $\frac{3}{p}$.

There are totally $\binom{\psi_G}{2}$, pairs of polynomials from L. Note that, $\mathcal{A}$ is allowed to make at most m queries we have. Thus, $\psi_G \le m + n + 2$. Then,

$$\Pr[\mathsf{Bad}] \le \binom{\psi_G}{2} \frac{3}{p} \le (m + n + 2)^2 \cdot \frac{3}{2p}.$$

Now, a simple argument shows that

$$\begin{aligned}
\Pr\left[\beta = \beta'\right] &= \Pr\left[\beta = \beta' | \neg\mathsf{Bad}\right] \Pr\left[\neg\mathsf{Bad}\right] + \Pr\left[\beta = \beta' | \mathsf{Bad}\right] \Pr\left[\mathsf{Bad}\right] \\
&\le \Pr\left[\beta = \beta' | \neg\mathsf{Bad}\right] \left(1 - \Pr\left[\mathsf{Bad}\right]\right) + \Pr\left[\mathsf{Bad}\right] \\
&= \frac{1}{2} + \frac{1}{2} \Pr\left[\mathsf{Bad}\right]
\end{aligned}$$

Also,

$$\Pr\left[\beta = \beta'\right] \ge \Pr\left[\beta = \beta' | \neg\mathsf{Bad}\right] \left(1 - \Pr\left[\mathsf{Bad}\right]\right) = \frac{1}{2} - \frac{1}{2} \Pr\left[\mathsf{Bad}\right]$$

. This two results were combined together to gives us

$$\mathsf{Adv}^{\mathsf{cpa}_{\mathcal{D}}}_{\mathcal{A},\mathsf{bed}_{\mathsf{anon}}} = \left| \Pr\left[\beta = \beta'\right] - \frac{1}{2} \right| \le \frac{\Pr\left[\mathsf{Bad}\right]}{2} \le (m + n + 2)^2 \cdot \frac{3}{4p}.$$

□

6.2.4 Message Indistinguishability from Unprivileged Users

Theorem 7. *Let there exists a* ppt *adversary $\mathcal{A}$ who can break the message indistinguishability from unprivileged users user of* $\mathsf{bed}_{\mathsf{anon}}$ *with the non-negligible advantage then it is possible to construct an efficient adversary $\mathcal{B}$ that has a non-negligible advantage in breaking* DDH *in $\mathbb{G}$ where $\mathcal{G} = (p, g, \mathbb{G}) \leftarrow \mathsf{PGen}(1^\lambda)$ such that $\mathbb{G} = \langle g \rangle$ or an efficient adversary $\mathcal{C}$ that has a non-negligible advantage in breaking the semantic security symmetric-key encryption* SE.

Proof. This proof is done via a sequence of games starting with the real construction Game_0. We assume X_i denotes the event the adversary winning Game_i.

Game_0**:** The Game_0 is the real construction. Here, the challenger samples $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and $u_i, v_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [n]$. It sets $\mathsf{pk} = (g, g^{\alpha}, g^{u_1+\alpha v_1}, g^{u_2+\alpha v_2}, \ldots, g^{u_n+\alpha v_n})$. For all secret key queries on $i \in [n]$, the adversary responds with (u_i, v_i). When the adversary gives distinct sets $S_0, S_1 \subset [n]$ of same cardinality k', the game samples $\beta \xleftarrow{\$} \{0,1\}$, $\kappa \xleftarrow{\$} \mathcal{K}$ and responds with the ciphertext $\mathsf{CT}_S = (\mathsf{ct}, H_0, H_1, c_1, \ldots, c'_k)$ where

$$\mathsf{ct} = \mathsf{E}(\kappa, M_\beta); \qquad H_0 = g^s; \qquad H_1 = g^{\alpha s}; \qquad c_i = g^{s(u_i+\alpha v_i)} \cdot \kappa.$$

The adversary finally returns $\beta' \in \{0,1\}$ as its guess.

Game_1**:** This game is nothing but a conceptual change of Game_0. We modify the way ciphertext is produced here. Precisely, we use h_0 and h_1 to construct $\{\omega_i\}_{i\in[k']}$. When the adversary gives distinct sets $S_0, S_1 \subset [n]$ of same cardinality k', the game samples $\beta \xleftarrow{\$} \{0,1\}$, $\kappa \xleftarrow{\$} \mathcal{K}$ and responds with the ciphertext $\mathsf{CT}_S = (\mathsf{ct}, H_0, H_1, c_1, \ldots, c'_k)$ where

$$\mathsf{ct} = \mathsf{E}(\kappa, M_\beta); \qquad H_0 = g^s; \qquad H_1 = g^{\alpha s}; \qquad c_i = H_0^{u_i} H_1^{v_i} \cdot \kappa.$$

Thus, $\Pr[X_1] = \Pr[X_0]$.

Game_2**:** In this game, we change the way h_1 is sampled. Precisely, we sample h_1 uniformly at random. When the adversary gives distinct sets $S_0, S_1 \subset [n]$ of same cardinality k', the game samples $\beta \xleftarrow{\$} \{0,1\}$, $\kappa \xleftarrow{\$} \mathcal{K}$ and responds with the ciphertext $\mathsf{CT}_S = (\mathsf{ct}, H_0, H_1, c_1, \ldots, c'_k)$ where

$$\mathsf{ct} = \mathsf{E}(\kappa, M_\beta); \qquad H_0 = g^s; \qquad H_1 \xleftarrow{\$} \mathbb{G}; \qquad c_i = H_0^{u_i} H_1^{v_i} \cdot \kappa.$$

Note that, any adversary that can distinguish between Game_1 and Game_2, can be used to solve the DDH problem.

Lemma 3. *There exists a* ppt *adversary such that* $|\Pr[X_2] - \Pr[X_1]| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{DDH}}(\lambda)$.

Proof. Given the DDH problem instance (g, g^{α}, g^s, T), we define $\mathcal{B}$ to choose $\alpha, u_1, v_1, \ldots, u_n, v_n \xleftarrow{\$} \mathbb{Z}_p$. $\mathcal{B}$ then computes $\mathsf{pk} = (g, g^{\alpha}, g^{u_1}(g^{\alpha})^{v_1}, \ldots, g^{u_n}(g^{\alpha})^{v_n})$. For all secret key queries on $i \in [n]$, the adversary responds with (u_i, v_i). When the adversary gives the target set $S \subset [n]$ of cardinality k', $\mathcal{B}$ samples $\beta \xleftarrow{\$} \{0,1\}$, $\kappa \xleftarrow{\$} \mathcal{K}$ and responds with the ciphertext $\mathsf{CT}_S = (\mathsf{ct}, H_0, H_1, c_1, \ldots, c'_k)$ where

$$\mathsf{ct} = \mathsf{E}(\kappa, M_\beta); \qquad H_0 = g^s; \qquad H_1 = g^c; \qquad c_i = H_0^{u_i} H_1^{v_i} \cdot \kappa.$$

If $T = g^{\alpha s}$, $\mathcal{B}$ simulates Game_1 whereas if $T = g^c$ for $c \xleftarrow{\$} \mathbb{Z}_p$, then $\mathcal{B}$ simulates Game_2. □

Game_3**:** In this game, we change the way pk is computed and then show that $\Pr[X_3] - \Pr[X_2] = 0$. To do so, first, consider the selective variant of these games, that is, Game_{2^*} and Game_{3^*}, which are as Game_2 and Game_3 except that the adversary has to commit to the target set S of size k' before it gets the public key pk. We basically first show that $\Pr[X_{3^*}] - \Pr[X_{2^*}] = 0$ and then use the complexity leveraging we argue $\Pr[X_3] - \Pr[X_2] = \binom{n}{k'}(\Pr[X_{3^*}] - \Pr[X_{2^*}]) = 0$.

We now argue $\Pr[X_{3^*}] - \Pr[X_{2^*}] = 0$. The challenger here samples $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and $u_i, v_i, \mu_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [n]$. It defines $u'_i = \begin{cases} u_i - \alpha\mu_i & \text{if } i \in S \\ u_i & \text{otherwise} \end{cases}$ and $v'_i = \begin{cases} v_i + \mu_i & \text{if } i \in S \\ v_i & \text{otherwise} \end{cases}$ and outputs $\mathsf{pk} = (g, g^{\alpha}, g^{u'_1}(g^{\alpha})^{v'_1}, \ldots, g^{u'_n}(g^{\alpha})^{v'_n})$. Observe that pk is exactly same as in Game_2. Precisely, $\{\mu_i\}_{i\in[n]}$ are not leaked from pk. For all secret key queries on $i \in [n]$, the adversary responds with (u_i, v_i).

When the adversary gives distinct messages M_0, M_1 of same size, the game samples $\beta \xleftarrow{\$} \{0,1\}$, $\kappa \xleftarrow{\$} \mathcal{K}$ and responds with the ciphertext CT_S on M_β where $\mathsf{CT}_S = (\mathsf{ct}, H_0, H_1, c_1, \ldots, c'_k)$ where

$$\mathsf{ct} = \mathsf{E}(\kappa, M_\beta); \qquad H_0 = g^s; \qquad H_1 = g^{(\alpha s+\delta)}; \qquad c_i = H_0^{u'_i} H_1^{v'_i} \cdot \kappa.$$

This is simply a conceptual change of the way pk and the challenge ciphertext is computed. Thus, $\Pr[X_{3^*}] - \Pr[X_{2^*}] = 0$.

Game$_4$**:** Here, we replace ct in the challenge ciphertext CT_S. Precisely, we return $\mathsf{ct} \leftarrow \mathsf{E}(\kappa, 0)$. Note that such a change is indistinguishable from Game$_3$ due to the semantic security of SE. Thus, $\Pr[X_4] - \Pr[X_3] \leq \mathsf{Adv}_{\mathcal{C}}^{\mathsf{SE}}(\lambda)$.

Now observe that, the ciphertext components $c_i = H_0^{u_i'} H_1^{v_i'} \cdot \kappa = H_0^{u_i} H_1^{v_i} g^{-\alpha s \mu_i} g^{\delta \mu_i + \alpha s \mu_i} \kappa = H_0^{u_i} H_1^{v_i} g^{\delta \mu_i} \kappa$ for all $i \in S$. Since, μ_i are uniformly random element, $g^{\delta \mu_i}$ are purely random element that hides κ for all $i \in S$. Thus, κ is a uniformly random element in the adversary's view. Thus, $\Pr[X_4] = 1/2$.

□

7 Adaptively Secure BED with Constant-size Ciphertext (bed$_{\mathcal{P}}$)

Our second broadcast encryption with dealership protocol (bed$_{\mathsf{anon}}$) presented in the last section, achieves adaptively secure message indistinguishability from unprivileged users. However, in bed$_{\mathsf{anon}}$ ciphertext size depends upon the privileged set size. This section presents our final construction of BED where we improve upon ciphertext length of bed$_{\mathsf{anon}}$ maintaining the same security guarantee. The final construction uses a prime order bilinear pairing group. This construction would further be referred to as bed$_{\mathcal{P}}$. We adapt the broadcast encryption technique of [RWZ12] for this construction. Note that, the broadcast encryption of [RWZ12] does not have anonymous security. Like bed$_{\mathcal{C}}$, we also manage to achieve the group privacy security (priv) under the standard DDH assumption.

7.1 Construction

It is defined by the following ppt algorithms.

- Setup$(1^\lambda, n)$:
  1. Run $\mathcal{PG} = (p, g, h, \mathbb{G}, \mathbb{H}, \mathbb{G}_{\mathrm{T}}, e) \xleftarrow{\$} \mathsf{PBGen}(1^\lambda)$.
  2. choose $\alpha, \gamma, \delta, c \xleftarrow{\$} \mathbb{Z}_p^*$, and set $f(x) = cx$.
  3. Set $u = g^\delta$ and $z = h^\delta$, define $u_i = u^{\alpha^i}$ for $i \in [n]$.
  4. Set $v = g^{\alpha\gamma}$, define $v_i = v^{\alpha^i}$ for $i \in [n]$.
  5. Set $\mathsf{msk} = (\alpha, \gamma, \delta)$.
  6. Publish $\mathsf{pk} = (g, h, u, v, g_1, \ldots, g_n, h_1, \ldots, h_n, u_1, \ldots, u_n, v_1, \ldots, v_n, f(x))$, where $g_i = g^{\alpha^i}$ and $h_i = h^{\alpha^i}$.
- KeyGen(msk, i):
  1. choose $r \xleftarrow{\$} \mathbb{Z}_p^*$, $w_i \xleftarrow{\$} \mathbb{H}$.
  2. Output $\mathsf{sk}_i = (\mathsf{sk}_{i,1}, \mathsf{sk}_{i,2}, \mathsf{sk}_{i,3}, \mathsf{sk}_{i,4}, \mathsf{sk}_{i,5})$ where,

$$\begin{aligned} \mathsf{sk}_{i,1} &= (w_i \cdot h^{r_i})^{\frac{1}{\alpha\gamma(\alpha - ID_i)}} & \mathsf{sk}_{i,2} &= r_i \\ \mathsf{sk}_{i,3} &= z & \mathsf{sk}_{i,5} &= (w_i, w_i^{\alpha}, \ldots, w_i^{\alpha^n}) . \\ \mathsf{sk}_{i,4} &= (z^{f(r_i)} \cdot w_i)^{\frac{1}{\alpha\gamma}} \end{aligned}$$

- GroupGen(pk, k, S):
  1. Suppose S is the set of users to who has enlisted to the dealer and let $|S| = k' \leq k$.
  2. Define $P_S(z) = \prod_{ID_i \in S} (z - ID_i)$.
  3. choose $s \xleftarrow{\$} \mathbb{Z}_p^*$ and output $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ where,

$$\begin{aligned} \omega_1 &= e(g, h)^{-s} & \omega_2 &= g^{-\alpha s} & & \\ \omega_4 &= v^{s P_S(\alpha)} & \omega_5 &= v^{\alpha^{n-k} \cdot s P_S(\alpha)} & \omega_3 &= e(u, h)^s . \end{aligned}$$

- Verify$(\mathsf{pk}, \mathit{\Gamma}_S, k)$: If $e(\omega_5, h^{\alpha^k}) = e(\omega_4, h^{\alpha^n})$, return 1. Otherwise, output 0.

- $\mathsf{Encrypt}(\mathsf{pk}, \mathit{\Gamma}_S, M)$: Parse $\mathit{\Gamma}_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$,
  1. choose $t \xleftarrow{\$} \mathbb{Z}_p^*$.
  2. Output $\mathsf{CT}_S = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3, \mathsf{ct}_4)$ where

$$\mathsf{ct}_1 = (\omega_1)^t \qquad \mathsf{ct}_2 = (\omega_2)^t$$
$$\mathsf{ct}_3 = M \cdot (\omega_3)^t \qquad \mathsf{ct}_4 = (\omega_4)^t.$$

- $\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_i, (S, \mathsf{CT}_S))$: Parse $\mathsf{CT}_S = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3, \mathsf{ct}_4)$ where,

$$\mathsf{ct}_1 = e(g, h)^{-t'} \qquad \mathsf{ct}_2 = g^{-\alpha t'}$$
$$\mathsf{ct}_3 = M \cdot e(u, h)^{t'} \qquad \mathsf{ct}_4 = v^{t' P_S(\alpha)}.$$

Correctness. Here $t' = st$.

$$\left(e(\mathsf{ct}_4, \mathsf{sk}_{i,1}) \cdot e(\mathsf{ct}_2, w_i h^{\mathsf{sk}_{i,2}})^{A_{i,S}(\alpha)}\right)^{\frac{1}{\prod_{j\in S}^{j\neq i}(-ID_j)}} \cdot \mathsf{ct}_1^{\mathsf{sk}_{i,2}} = e(g, w_i)^{st}$$

$$\left(e(\mathsf{ct}_4, \mathsf{sk}_{i,4}) \cdot e(\mathsf{ct}_2, z^{\mathsf{sk}_{i,2}} w_i)^{B_{i,S}(\alpha)}\right)^{\frac{1}{\prod_{j\in S}(-ID_j)}} = e(g, z^{f(r_i)} h_i)^{st}$$

$$\left(\frac{e(g, z^{f(r_i)} w_i)^{st}}{e(g, w_i)^{st}}\right)^{\frac{1}{f(\mathsf{sk}_{i,2})}} = e(g, z)^{st} = e(g, h)^{\delta st} = e(g^\delta, h)^{st} = e(u, h)^{st}$$

$$M = \frac{\mathsf{ct}_3}{e(u, h)^{st}}.$$

Where, $A_{i,S}(\alpha) = \dfrac{1}{\alpha}\left(\prod_{j\in S}^{j\neq i}(\alpha - ID_j) - \prod_{j\in S}^{j\neq i}(-ID_j)\right)$ and $B_{i,S}(\alpha) = \dfrac{1}{\alpha}\left(\prod_{i\in S}(\alpha - ID_i) - \prod_{i\in S}(-ID_i)\right)$.

7.2 Security

We prove the above construction achieves security in all the four security models defined in section 3.3.

7.2.1 Group Privacy

Theorem 8. *If there exists a* ppt *adversary $\mathcal{A}$ breaking the group privacy of* $\mathsf{bed}_{\mathcal{P}}$ *with a non-negligible advantage, then there exists a* ppt *adversary $\mathcal{B}$ which has a non-negligible advantage in solving DDH problem in $\mathbb{G}$ where $\mathcal{PG} = (p, g, h, \mathbb{G}, \mathbb{H}, \mathbb{G}_T) \leftarrow \mathsf{PBGen}(1^\lambda)$ such that $\mathbb{G} = \langle g \rangle$.*

Proof. Let $\mathcal{A}$ be a ppt adversary breaking priv security of $\mathsf{bed}_{\mathcal{P}}$, we want to construct a ppt adversary $\mathcal{B}$ for DDH that uses $\mathcal{A}$ as a subroutine. Given a DDH problem instance (g, h, g^a, g^b, Z) for $a, b \xleftarrow{\$} \mathbb{Z}_p$, simulate $\mathcal{A}$ as following:

- Sample $\alpha, \delta, c \xleftarrow{\$} \mathbb{Z}_p$.
- Implicitly set $\gamma = a$ and $s = b$.
- Set $u = g^\delta$, $z = h^\delta$, $v = (g^a)^\alpha$, and $f(x) = cx$ and publish,

$$\mathsf{pk} = (g, h, u, g^\alpha, \ldots, g^{\alpha^n}, h^\alpha, \ldots, h^{\alpha^n}, u^\alpha, \ldots, u^{\alpha^n}, v^\alpha, \ldots, v^{\alpha^n}, f(x)).$$

- $\mathcal{A}$ output two sets (S_0, S_1) such that $|S_0| = |S_1| < n$.
- Given S_0 and S_1 of same size k (say), choose $\beta \xleftarrow{\$} \{0, 1\}$, and return the token $\mathit{\Gamma}_{S_\beta} = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ as following:
  - $\omega_1 = e(g^b, h)^{-1} = e(g, h)^{-b} = e(g, u)^{-s}$.
  - $\omega_2 = (g^b)^{-\alpha} = g^{-\alpha s}$.
  - $\omega_3 = e(u^b, h) = e(u, h)^b = e(u, h)^s$.
  - $\omega_4 = (Z^\alpha)^{P_{S_\beta}(\alpha)}$ now if $Z = g^{ab}$ then $\omega_4 = (g^a)^\alpha)^{bP_{S_\beta}(\alpha)} = v^{sP_{S_\beta}(\alpha)}$
  - $\omega_5 = ((Z^\alpha)^\alpha)^{\alpha^{n-k}.P_{S_\beta}(\alpha)}$ now if $Z = g^{ab}$ then $\omega_5 = (g^a)^\alpha)^{\alpha^{n-k}\cdot sP_{S_\beta}(\alpha)} = v^{\alpha^{n-k}\cdot sP_{S_\beta}(\alpha)}$.

- $\mathcal{A}$ outputs $\hat{\beta}$.
- If $\hat{\beta} = \beta$, then output 1 else 0.

As α, δ, c are chosen randomly, the public key provided by $\mathcal{B}$ has identical distribution to that of the original construction. So when the input of $\mathcal{B}$ is g^{ab} then Γ_{S_β} is a valid group token to $\mathcal{A}$ and if Z is a random element of $\mathbb{G}$ then (ω_4, ω_5) are just two random elements of $\mathbb{G}$.

Now, the advantage of $\mathcal{B}$ in the DDH game (we denote $\mathcal{B}^{\mathsf{DDH}} \Rightarrow 1$ as $\mathcal{B}$ given the DDH instance outputs 1) is the same as the adversary of $\mathcal{A}$ guessing b in priv game other than guessing randomly. So,

$$\begin{aligned}
\Pr\left[\mathcal{B}^{\mathsf{DDH}} \Rightarrow 1\right] &\geq \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_\mathcal{P}} \Rightarrow 1\right] \\
\Pr\left[\mathcal{B}^{\mathsf{DDH}} \Rightarrow 1\right] &\geq \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_\mathcal{P}} \Rightarrow 1 | Z = g^{ab}\right] \Pr\left[Z = g^{ab}\right] \\
&\quad + \sum_{p-1} \Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_\mathcal{P}} \Rightarrow 1 | Z \xleftarrow{\$} \mathbb{G}\right] \Pr\left[Z \xleftarrow{\$} \mathbb{G}\right] \\
\frac{1}{p}\Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_\mathcal{P}} \Rightarrow 1 | Z = g^{ab}\right] &\leq \left|\Pr\left[\mathcal{B}^{\mathsf{DDH}} \Rightarrow 1\right] - \frac{p-1}{p}\Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_\mathcal{P}} \Rightarrow 1 | Z \xleftarrow{\$} \mathbb{G}\right]\right| \\
\frac{1}{p}\Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_\mathcal{P}} \Rightarrow 1 | Z = g^{ab}\right] &\leq \left|\Pr\left[\mathcal{B}^{\mathsf{DDH}} \Rightarrow 1\right] - \frac{p-1}{2p}\right| \\
\frac{1}{p}\Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_\mathcal{P}} \Rightarrow 1 | Z = g^{ab}\right] &\leq \Big|\Pr\left[\mathcal{B}^{\mathsf{DDH}} \Rightarrow 1 | Z = g^{ab}\right] \Pr\left[Z = g^{ab}\right] \\
&\quad \left. + \sum_{p-1} \Pr\left[\mathcal{B}^{\mathsf{DDH}} \Rightarrow 1 | Z \xleftarrow{\$} \mathbb{G}\right] \Pr\left[Z \xleftarrow{\$} \mathbb{G}\right] - \frac{p-1}{2p}\right| \\
\Pr\left[\mathcal{A}^{\mathsf{priv}}_{\mathsf{bed}_\mathcal{P}} \Rightarrow 1 | Z = g^{ab}\right] &\leq \left|\Pr\left[\mathcal{B}^{\mathsf{DDH}} \Rightarrow 1 | Z = g^{ab}\right] + (p-1)\left(\frac{1}{2} - \frac{1}{2}\right)\right| \\
\mathsf{Adv}^{\mathsf{priv}}_{\mathcal{A},\mathsf{bed}_\mathcal{P}} &\leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{B}}
\end{aligned}$$

□

7.2.2 Maximum Users Accountability Let us consider three random encodings function $\sigma_G : \mathbb{G} \to \{0,1\}^{m_G}$, $\sigma_H : \mathbb{H} \to \{0,1\}^{m_H}$ and $\sigma_T : \mathbb{G}_T \to \{0,1\}^{m_T}$, where w.l.o.g. $m_G \leq m_H \leq m_T$.

Theorem 9. *Let $\mathcal{A}$ be any generic group adversary for the maximum user accountability security of* $\mathsf{bed}_\mathcal{P}$. *$\mathcal{A}$ make queries to oracles computing the group actions in $\mathbb{G}$, $\mathbb{H}$, $\mathbb{G}_T$ and the bilinear map e. Let m be a bound on the total number of group elements $\mathcal{A}$ receives. Then we have that the advantage of $\mathcal{A}$ in the* maximum user accountability *security game of* $\mathsf{bed}_\mathcal{P}$ *is bounded by $\mathcal{O}(m^2/p)$.*

Proof. Let $\mathcal{C}$ denote an algorithm that simulates the generic bilinear group for $\mathcal{A}$. To answer queries from $\mathcal{A}$, $\mathcal{C}$ maintains three lists,

$$\begin{aligned}
L_G &= \{(f_{G,i}, \sigma_{G,i}) : i \in [0, \psi_G - 1]\} \\
L_H &= \{(f_{H,i}, \sigma_{H,i}) : i \in [0, \psi_H - 1]\} \\
L_T &= \{(f_{T,i}, \sigma_{T,i}) : i \in [0, \psi_T - 1]\}
\end{aligned}$$

such that at each step ψ of the game, the relation $\psi_G + \psi_H + \psi_T = \psi + 4n + 4$ holds. Here $f_{*,*}$ are multivariate polynomials over 5 variables α, γ, δ, r, s and $\sigma_{b,i}$ are strings from $\{0,1\}^{m_b}$ where $b \in \{G, H, T\}$. At the beginning of the game i.e., $\psi = 0$, the lists are initialized by setting $\psi_G = 3(n+1)$, $\psi_H = (n+1)$ and $\psi_T = 0$. The polynomials 1, α, …, α^n, δ, $\delta\alpha$, …, $\delta\alpha^n$, $\gamma\alpha$, …, $\gamma\alpha^{n+1}$ are assigned to $f_{G,0}$, $f_{G,1}$, …, $f_{G,n}$, $f_{G,n+1}$, …, $f_{G,2n+1}$, $f_{G,2n+2}$, …, $f_{G,3n+2}$; 1, α, …, α^n are assigned to $f_{H,0}$, $f_{H,1}$, …, $f_{H,n}$. For each of these polynomials, the associated encodings are strings uniformly chosen from $\{0,1\}^{m_b}$ without repetition for $b \in \{G, H, T\}$. We assume $\mathcal{A}$ has to query the oracles for any group element or pairing computation. Naturally $\mathcal{C}$ can obtain the index of a given string $\sigma_{b,i}$ in the list L_b. The oracles are simulated as follows.

Group Actions in $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{G}_T$. We describe this for the group $\mathbb{G}$. We note that the group actions in $\mathbb{H}$ and $\mathbb{G}_T$ are simulated similarly. If $\mathcal{A}$ submits two strings $\sigma_{G,i}$ and $\sigma_{G,j}$ and a sign bit indicating addition or subtraction. $\mathcal{C}$ first finds $f_{G,i}$ and $f_{G,j}$ corresponding to $\sigma_{G,i}$ and $\sigma_{G,j}$ respectively in L_G and computes $f_{G,\psi_G} = f_{G,i} \pm f_{G,j}$. If there exists an index $k \in [0, \psi_G - 1]$, such that $f_{G,\psi_G} = f_{G,k}$, $\mathcal{C}$ sets $\sigma_{G,\psi_G} = \sigma_{G,k}$; otherwise $\mathcal{C}$ sets $\sigma_{G,\psi_G} \xleftarrow{\$} \{0,1\}^{m_G} \setminus \{\sigma_{G,0}, \sigma_{G,1}, \ldots, \sigma_{G,\psi_G - 1}\}$, add $(f_{G,\psi_G}, \sigma_{G,\psi_G})$ to L_G, returns σ_{G,ψ_G} to $\mathcal{A}$ and increments ψ_G by one.

Bilinear Map. If $\mathcal{A}$ submits two strings $\sigma_{G,i}$ and $\sigma_{H,j}$, $\mathcal{C}$ first finds $f_{G,i}$ in L_G corresponding to $\sigma_{H,i}$ and $f_{H,j}$ in L_H corresponding to $\sigma_{H,j}$ respectively and computes $f_{T,\psi_T} = f_{G,i} \pm f_{H,j}$. If there exists an index $k \in [0, \psi_T - 1]$, such that $f_{T,\psi_T} = f_{T,k}$, $\mathcal{C}$ sets $\sigma_{T,\psi_T} = \sigma_{T,k}$; otherwise $\mathcal{C}$ sets $\sigma_{T,\psi_T} \xleftarrow{\$} \{0,1\}^{m_T} \setminus \{\sigma_{T,0}, \sigma_{T,1}, \ldots, \sigma_{T,\psi_T - 1}\}$ and add $(f_{T,\psi_T}, \sigma_{T,\psi_T})$ to L_T and returns σ_{T,ψ_T} to $\mathcal{A}$ and increments ψ_T by one.

At this point, $\mathcal{A}$ aborts by producing a challenge $(k, \sigma'_G, \sigma''_G, \sigma'''_G, \sigma'_T, \sigma''_T)$ such that (f'_G, σ'_G), (f''_G, σ''_G), $(f'''_G, \sigma'''_G) \in L_G$, and $(f'_T, \sigma'_T), (f''_T, \sigma''_T) \in L_T$. Observe that, $(f'_G, f''_G, f'''_G,\ f'_T, f''_T)$ are polynomials of α, γ, δ and s where α, γ, δ will be sampled by $\mathcal{C}$ and s by $\mathcal{A}$. The generic group model ensures that $\mathcal{C}$ can verify $f'_T = f'_G \cdot \alpha^{-1}$, $f''_T = f'_G \cdot \delta^{-1}\alpha$, $f''_T = (f'_T)^{-1}\delta\ f'''_G = f''_G \cdot \alpha^{n-k}$ and $f''_G \cdot \gamma^{-1} \cdot {f'_T}^{-1} \in \mathsf{Span}(f_{G,0}, f_{G,1}, \ldots, f_{G,n})$.

Let $\mathbf{v} = (\alpha, \gamma, \delta, r)$ denote the vector consisting of variables over which the polynomials are defined. Now the simulator chooses at random $\alpha^*, \gamma^*, \delta^*, r^* \xleftarrow{\$} \mathbb{Z}_p$. Let $\mathbf{v}^* = (\alpha^*, \gamma^*, \delta^*, r^*)$. $\mathcal{C}$ assigns $\mathbf{v}^*$ to the variables of $\mathbf{v}$. The simulation provided by $\mathcal{C}$ is perfect unless for some $i, j \le \psi_b$ any of the following holds.

1. $f_{G,i}(\mathbf{v}^*) - f_{G,j}(\mathbf{v}^*) = 0$ for some $i \neq j$ but $f_{G,i} \neq f_{G,j}$.
2. $f_{H,i}(\mathbf{v}^*) - f_{H,j}(\mathbf{v}^*) = 0$ for some $i \neq j$ but $f_{H,i} \neq f_{H,j}$.
3. $f_{T,i}(\mathbf{v}^*) - f_{T,j}(\mathbf{v}^*) = 0$ for some $i \neq j$ but $f_{T,i} \neq f_{T,j}$.

We use Bad to denote the event that at least one of the above holds, and we shall try to bound the probability of Bad. If Bad does not happen, then the simulation was perfect. Assume $\mathcal{A}$ has generated his challenge for the set S, with $|S| > k$. Then in case Bad does not happen, $\mathcal{A}$ has no advantage in guessing $\mathit{\Gamma}_S$ over a random guess. In the polynomial f'''_G, the highest possible degree of α is $n+1$ if $|S| \le k$. If $\mathcal{A}$ tries to simulate any group token where $|S| > k$ then in the polynomial f'''_G the highest degree of α is greater than $n+1$. Notice that α^{n+i} for some $i \ge 1$ is independent of $(1, \alpha, \ldots, \alpha^n)$. Also $\mathcal{A}$ does not have access to $g^{\alpha^{n+1+i}}$, which is outside the span of pk. Thus, it has one and only option is to guess it. Probability that it guesses f'''_G having no info about $g^{\alpha^{n+1+i}}$ is negligible.

Now, we need to bound the probability of Bad. A result by Schwartz [Sch80] would be used here. The result states, for an n-variate polynomial $F(x_1, \ldots, x_n) \in \mathbb{Z}_p[X_1, \ldots, X_n]$ of degree d, a random assignment $x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p$ make the polynomial F evaluate to zero with probability at most d/p. For fixed i, j, $f_{G,i} - f_{G,j}$ is a polynomial of degree at most $n+2$, hence zero at random $\mathbf{v}^*$ with probability at most $\frac{n+2}{p}$. For fixed i, j, $f_{H,i} - f_{H,j}$ is a polynomial of degree at most n, hence zero at random $\mathbf{v}^*$ with probability at most $\frac{n}{p}$. For fixed i, j, $f_{T,i} - f_{T,j}$ is a polynomial of degree at most $n(n+2)$, hence zero at random $\mathbf{v}^*$ with probability at most $\frac{n(n+2)}{p}$. There are totally $\binom{\psi_G}{2}$, $\binom{\psi_H}{2}$, $\binom{\psi_T}{2}$ pairs of polynomials from L_G, L_H and L_T respectively. Note that, $\mathcal{A}$ is allowed to make at most m queries. Thus we have, $\psi_G + \psi_H + \psi_T \le m + 4n + 4$. Then,

$$\Pr[\mathsf{Bad}] \le \binom{\psi_G}{2}(n+2)/p + \binom{\psi_H}{2} n/p + \binom{\psi_T}{2} n(n+2)/p \le (m+4n+4)^2 \cdot \frac{n^2+4n+2}{2p}.$$

So $\Pr[\neg\mathsf{Bad}] = 1 - (m+4n+4)^2 \cdot \frac{n^2+4n+2}{2p}$ and $\Pr[\mathcal{A}\text{ wins}|\neg\mathsf{Bad}] = \frac{2}{2p-(m+4n+4)^2 \cdot (n^2+4n+2)}$.

Now

$$\Pr[\mathcal{A}\text{ wins}] = \Pr[\mathcal{A}\text{ wins}|\neg\mathsf{Bad}]\Pr[\neg\mathsf{Bad}] + \Pr[\mathcal{A}\text{ wins}|\mathsf{Bad}]\Pr[\mathsf{Bad}].$$

So if Bad does not happen then $\mathcal{A}$ "knows" nothing about those possible values where any two $f_{b,i}(x) = f_{b,j}(x)$ happen for $1 \le i < j \le \psi_b$. Considering all this together probability that $\mathcal{A}$ wins is bounded by $\mathcal{O}(m^2/p)$.

□

7.2.3 Message Indistinguishability from Dealer

Let us consider three random encodings function $\sigma_G : \mathbb{G} \to \{0,1\}^{m_G}$, $\sigma_H : \mathbb{H} \to \{0,1\}^{m_H}$ and $\sigma_T : \mathbb{G}_T \to \{0,1\}^{m_T}$ where w.l.o.g. $m_G \leq m_H \leq m_T$.

Theorem 10. *Let $\mathcal{A}$ be a* ppt *adversary against message indistinguishability from dealer security of $\mathsf{bed}_\mathcal{P}$ in the generic group model. $\mathcal{A}$ make queries to the oracles computing the group actions in $\mathbb{G}$, $\mathbb{H}$, $\mathbb{G}_T$ and the bilinear map e. Let m be a bound on the total number of group elements $\mathcal{A}$ receives from queries and n be any natural number. Then we have that the advantage of $\mathcal{A}$ in the* message indistinguishability from dealer *security game of $\mathsf{bed}_\mathcal{P}$ is at most*

$$\mathsf{Adv}^{\mathsf{cpa}_\mathcal{D}}_{\mathcal{A},\mathsf{bed}_\mathcal{P}} \leq (m + 4n + 4)^2 \cdot \frac{n^2 + 4n + 2}{4p}.$$

Proof. Let $\mathcal{C}$ be the algorithm that simulates the generic bilinear group for $\mathcal{A}$. To answer oracle queries of $\mathcal{A}$, $\mathcal{C}$ maintains three lists,

$$L_G = \{(f_{G,i}, \sigma_{G,i}) : i \in [0, \psi_G - 1]\}$$
$$L_H = \{(f_{H,i}, \sigma_{H,i}) : i \in [0, \psi_H - 1]\}$$
$$L_T = \{(f_{T,i}, \sigma_{T,i}) : i \in [0, \psi_T - 1]\}$$

such that at each step ψ of the game, the relation $\psi_G + \psi_H + \psi_T = \psi + 4n + 4$ holds. Here $f_{*,*}$ are multivariate polynomials over 6 variables α, γ, δ, s, y_0, y_1 and $\sigma_{b,i}$ are strings from $\{0,1\}^{m_b}$ where $b \in \{G, H, T\}$. At the beginning of the game i.e., $\psi = 0$, the lists are initialized by setting $\psi_G = 3(n+1)$, $\psi_H = (n+1)$ and $\psi_T = 0$. The polynomials 1, α, $\ldots$, α^n, δ, $\delta\alpha$, $\ldots$, $\delta\alpha^n$, $\gamma\alpha$, $\ldots$, $\gamma\alpha^{n+1}$ are assigned to $f_{G,0}$, $f_{G,1}$, $\ldots$, $f_{G,n}$, $f_{G,n+1}$, $\ldots$, $f_{G,2n+1}$, $f_{G,2n+2}$, $\ldots$, $f_{G,3n+2}$; 1, α, $\ldots$, α^n are assigned to $f_{H,0}$, $f_{H,1}$, $\ldots$, $f_{H,n}$. For these polynomials, the associated encodings are strings uniformly chosen from $\{0,1\}^{m_b}$ without repetition for $b \in \{G, H, T\}$. We assume that $\mathcal{A}$ queries the oracles on strings previously obtained from $\mathcal{C}$ and naturally $\mathcal{C}$ can obtain the index of a given string $\sigma_{b,i}$ in the list L_b. The oracles are simulated as follows.

Group Actions in $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{G}_T$. We describe this for the group $\mathbb{G}$. We note that the group actions in $\mathbb{H}$ and $\mathbb{G}_T$ are simulated similarly. If $\mathcal{A}$ submit two strings $\sigma_{G,i}$ and $\sigma_{G,j}$ and a sign bit indicating addition or subtraction. $\mathcal{C}$ first finds $f_{G,i}$ and $f_{G,j}$ corresponding to $\sigma_{G,i}$ and $\sigma_{G,j}$ respectively in L_G and computes $f_{G,\psi_G} = f_{G,i} \pm f_{G,j}$. If there exists an index $k \in [0, \psi_G - 1]$, such that $f_{G,\psi_G} = f_{G,k}$, $\mathcal{C}$ sets $\sigma_{G,\psi_G} = \sigma_{G,k}$; otherwise $\mathcal{C}$ sets $\sigma_{G,\psi_G} \xleftarrow{\$} \{0,1\}^{m_G} \setminus \{\sigma_{G,0}, \sigma_{G,1}, \ldots, \sigma_{G,\psi_G - 1}\}$ and add $(f_{G,\psi_G}, \sigma_{G,\psi_G})$ to L_G and returns σ_{G,ψ_G} to $\mathcal{A}$ and increments ψ_G by one.

Bilinear Map. If $\mathcal{A}$ submit two strings $\sigma_{G,i}$ and $\sigma_{H,j}$, $\mathcal{C}$ first finds $f_{G,i}$ in L_G corresponding to $\sigma_{G,i}$ and $f_{H,j}$ in L_H corresponding to $\sigma_{H,j}$ respectively and computes $f_{T,\delta_T} = f_{G,i} \pm f_{H,j}$. If there exists an index $k \in [0, \psi_T - 1]$, such that $f_{T,\psi_T} = f_{T,k}$, $\mathcal{C}$ sets $\sigma_{T,\psi_T} = \sigma_{T,k}$; otherwise $\mathcal{C}$ sets $\sigma_{T,\psi_T} \xleftarrow{\$} \{0,1\}^{m_T} \setminus \{\sigma_{T,0}, \sigma_{T,1}, \ldots, \sigma_{T,\psi_T - 1}\}$ and add $(f_{T,\psi_T}, \sigma_{T,\psi_T})$ to L_T and returns σ_{T,ψ_T} to $\mathcal{A}$ and increments ψ_T by one.

At this point, $\mathcal{A}$ produces a challenge $(k, \sigma'_G, \sigma''_G, \sigma'''_G, \sigma'_T, \sigma''_T)$ such that $(f'_G, \sigma'_G), (f''_G, \sigma''_G), (f'''_G, \sigma'''_G) \in L_G$, and $(f'_T, \sigma'_T), (f''_T, \sigma''_T) \in L_T$. Precisely, $(\sigma'_G, \sigma''_G, \sigma'''_G, \sigma'_T, \sigma''_T) = (s\alpha, sP_S(\alpha), \alpha^{n-k} sP_S(\alpha), s, s\delta)$. Observe that, $(f'_G, f''_G, f'''_G, f'_T, f''_T)$ are polynomials of α, γ, δ and s where α, γ, δ will be sampled by $\mathcal{C}$ and s will be sampled by $\mathcal{A}$. The generic group model ensures that $\mathcal{C}$ can verify $f'_T = f'_G \cdot \alpha^{-1}$, $f''_T = f'_G \cdot \delta^{-1}\alpha$, $f''_T = (f'_T)^{-\alpha}\delta$ $f'''_G = f''_G \cdot \alpha^{n-k}$ and $f''_G \cdot \gamma^{-1} \cdot {f'_T}^{-1} \in \mathsf{Span}(f_{G,0}, f_{G,1}, \ldots, f_{G,n})$. Finally, $\mathcal{C}$ samples $\beta \xleftarrow{\$} \{0,1\}$ and sets $\hat{f}'_G = f'_G \cdot y_0$, $\hat{f}''_G = f''_G \cdot y_0$, and $\hat{f}'_T = f'_T \cdot y_\beta$, $\hat{f}''_T = f''_T \cdot y_0$. $\mathcal{C}$ then adds $(\hat{f}'_G, \hat{\sigma}'_G), (\hat{f}''_G, \hat{\sigma}''_G) \in L_G$ and $(\hat{f}'_T, \hat{\sigma}'_T), (\hat{f}''_T, \hat{\sigma}''_T) \in L_T$ following the above rules on group actions. At this point, $\mathcal{C}$ returns $(\hat{\sigma}'_G, \hat{\sigma}''_G, \hat{\sigma}'_T, \hat{\sigma}''_T)$ to $\mathcal{A}$ who returns β' and abort. Let $\mathbf{v} = (\alpha, \gamma, \delta, y_0, y_1)$ denotes the vector consisting of variables over which the polynomials are defined. Now the simulator chooses at random $\alpha^*, \gamma^*, \delta^*, y_0^*, y_1^* \xleftarrow{\$} \mathbb{Z}_p$. Let $\mathbf{v}^* = (\alpha^*, \gamma^*, \delta^*, y_0^*, y_1^*)$. $\mathcal{C}$ assigns $\mathbf{v}^*$ to the variables of $\mathbf{v}$. The simulation provided by $\mathcal{C}$ is perfect unless for some $i, j \leq \psi_b$ any of the following holds.

1. $f_{G,i}(\mathbf{v}^*) - f_{G,j}(\mathbf{v}^*) = 0$ for some $i \neq j$ but $f_{G,i} \neq f_{G,j}$.

2. $f_{H,i}(\mathbf{v}^*) - f_{H,j}(\mathbf{v}^*) = 0$ for some $i \neq j$ but $f_{H,i} \neq f_{H,j}$.
3. $f_{T,i}(\mathbf{v}^*) - f_{T,j}(\mathbf{v}^*) = 0$ for some $i \neq j$ but $f_{T,i} \neq f_{T,j}$.

We use Bad to denote the event that at least one of the above holds, and we'll try to bound the probability of Bad. If Bad does not happen, then the simulation was perfect. So if Bad, does not happen, $\mathcal{A}$ has no advantage in guessing β' over a random guess. Precisely, $\Pr[\beta = \beta' | \neg\mathsf{Bad}] = 1/2$. To see this, observe that all variables except y_β and $y_{1-\beta}$ are independent of the bit β. Assume $y_\beta = r$ and recall $\mathcal{A}$ has access to all the lists (L_G, L_H, L_T) and gets $(\hat{\sigma}'_G, \hat{\sigma}''_G, \hat{\sigma}'_T, \hat{\sigma}''_T)$ as its challenge where $(\hat{f}'_G, \hat{\sigma}'_G), (\hat{f}''_G, \hat{\sigma}''_G) \in L_G$ and $(\hat{f}'_T, \hat{\sigma}'_T), (\hat{f}''_T, \hat{\sigma}''_T) \in L_T$. Observe that, $\hat{f}'_G, \hat{f}''_G, \hat{f}'_T, \hat{f}''_T$ are respectively $\alpha^* y_0^* s$, $\gamma^* y_0^* s P_S(\alpha^*)$, $y_0^* s$ and $\delta^* y_\beta^* s$. Where $P_S(\alpha)$ is a polynomial of degree at most n. It is clear that $\hat{f}'_T$ is a two-degree polynomial and $\hat{f}''_T$ is a three-degree polynomial defined in the group $\mathbb{G}_T$ but both $\hat{f}'_G$ and $\hat{f}''_G$ are polynomials of degree three or higher. To compute such $\hat{f}'_T$ and $\hat{f}''_T$, we mention that $\mathcal{A}$ can use the polynomial lists L_G and L_H and challenge polynomials $\hat{f}'_G$ and $\hat{f}''_G$. Indeed, composing $\hat{f}'_G$ and $\hat{f}''_G$ with L_G and L_H results in polynomials of three or higher degree. Those three-degree polynomials which can only be computed does not involve $\delta^* y_0^* s$. Thus, the best $\mathcal{A}$ can do is to output its guess β' at random and subsequently $\Pr[\beta = \beta' : \neg\mathsf{Bad}] = 1/2$.

Now, we show that the probability that Bad happens is negligible. This is where we utilize the result on random assignment of polynomial due to Schwartz [Sch80]. Roughly speaking, the result states that for an n-variate polynomial $F(x_1, \ldots, x_n) \in \mathbb{Z}_p[X_1, \ldots, X_n]$ of degree d, a random assignment $x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p$ make the polynomial F evaluate to zero with probability at most d/p. For fixed i, j, $f_{G,i} - f_{G,j}$ is a polynomial of degree at most $n+2$, hence zero at random $\mathbf{v}^*$ with probability at most $\frac{n+2}{p}$. For fixed i, j, $f_{H,i} - f_{H,j}$ is a polynomial of degree at most n, hence zero at random $\mathbf{v}^*$ with probability at most $\frac{n}{p}$. For fixed i, j, $f_{T,i} - f_{T,j}$ is a polynomial of degree at most $n(n+2)$, hence zero at random $\mathbf{v}^*$ with probability at most $\frac{n(n+2)}{p}$. There are totally $\binom{\psi_G}{2}$, $\binom{\psi_H}{2}$, $\binom{\psi_T}{2}$ pairs of polynomials from L_G, L_H and L_T respectively. Note that, $\mathcal{A}$ is allowed to make at most m queries we have. Thus, $\psi_G + \psi_H + \psi_T \leq m + 4n + 4$. Then,

$$\Pr[\mathsf{Bad}] \leq \binom{\psi_G}{2} (n+2)/p + \binom{\psi_H}{2} n/p + \binom{\psi_T}{2} n(n+2)/p \leq (m + 4n + 4)^2 \cdot \frac{n^2 + 4n + 2}{2p}.$$

Now, a simple argument shows that,

$$\begin{aligned}
\Pr[\beta = \beta'] &= \Pr[\beta = \beta' | \neg\mathsf{Bad}] \Pr[\neg\mathsf{Bad}] + \Pr[\beta = \beta' | \mathsf{Bad}] \Pr[\mathsf{Bad}] \\
&\leq \Pr[\beta = \beta' | \neg\mathsf{Bad}] (1 - \Pr[\mathsf{Bad}]) + \Pr[\mathsf{Bad}] \\
&= \frac{1}{2} + \frac{1}{2} \Pr[\mathsf{Bad}]
\end{aligned}$$

Also,

$$\Pr[\beta = \beta'] \geq \Pr[\beta = \beta' | \neg\mathsf{Bad}] (1 - \Pr[\mathsf{Bad}]) = \frac{1}{2} - \frac{1}{2} \Pr[\mathsf{Bad}]$$

. This two results were combined together to gives us

$$\mathsf{Adv}^{\mathsf{cpa}_{\mathcal{D}}}_{\mathcal{A}, \mathsf{bed}_{\mathcal{P}}} = \left| \Pr[\beta = \beta'] - \frac{1}{2} \right| \leq \frac{\Pr[\mathsf{Bad}]}{2} \leq (m + 4n + 4)^2 \cdot \frac{n^2 + 4n + 2}{4p}.$$

□

7.2.4 Message Indistinguishability from Unprivileged Users

The message indistinguishability from an unprivileged user is the first and foremost security guarantee we need from any variant of BE. Below we describe the $\mathsf{cpa}_{\mathcal{U}}$ security of our construction.

Theorem 11. *Let $\mathcal{A}$ be a* ppt *adversary breaking the* $\mathsf{cpa}_{\mathcal{U}}$ *security of* $\mathsf{bed}_{\mathcal{P}}$ *with non-negligible advantage then there exists a* ppt *adversary $\mathcal{B}$ that breaks the* waABDHE *assumption in $\mathcal{PG} = (p, g, h, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) \xleftarrow{\$}$* PBGen *with non-negligible advantage.*

Proof. Let $\mathcal{A}$ be a $\mathsf{cpa}_{\mathcal{U}}$ adversary as described in section 3.3.4 breaks the message indistinguishability from an unprivileged user of $\mathsf{bed}_{\mathcal{P}}$. We would argue that, if such an adversary $\mathcal{A}$ exists, then we can construct another ppt adversary $\mathcal{B}$ which will solve q-$wa\mathsf{ABDHE}$ problem.
Let $\mathcal{B}$ has the problem instance,

$$(g, h, g^{\lambda}, h^{\lambda}, g^{\alpha}, \ldots, g^{\alpha^q}, h^{\alpha}, \ldots, h^{\alpha^q}, g^{\lambda\alpha^{q+2}}, \ldots, g^{\lambda\alpha^{2q}}, h^{\lambda\alpha^{q+2}}, \ldots, h^{\lambda\alpha^{2q}}, Z) \in \mathbb{G}^{2q+1} \times \mathbb{H}^{2q+1} \times \mathbb{G}_T$$

problem is to decide $Z = e(g^{\lambda}, h)^{\alpha^{q+1}}$ or $Z = e(g, h^{\lambda})^{\alpha^{q+1}}$ or a random element from $\mathbb{G}_T$, where $g \xleftarrow{\$} \mathbb{G}, h \xleftarrow{\$} \mathbb{H}$, $\alpha, \lambda \xleftarrow{\$} \mathbb{Z}_p^*$.
For convenience we would refer $g^{\lambda} = g'$ and $h^{\lambda} = h'$. Simulation for $\mathcal{A}$ is as follows.
Setup: We would assume that in $\mathsf{bed}_{\mathcal{P}}$, the maximal size of the set of receiver is n and $q \geq 2n$.

- Sample $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$.
- Set $P(x) = \sum\limits_{i=0}^{n-1} \beta_{0,i} x^i$; $Q(x) = xP(x) + \beta_0$ and $f(x) = \frac{1}{\beta_0} x$ where $\beta_{0,i}, \beta_0 \xleftarrow{\$} \mathbb{Z}_p^*$.
- Also set $u = g^{Q(\alpha)}$, $z = h^{Q(\alpha)}$, $v = g^{\alpha\gamma}$ and $u^{\alpha^i} = g^{\alpha^i Q(\alpha)}$, $v^{\alpha^i} = g^{\alpha^{i+1}\gamma}$ for $i \in [n]$.

Publish $\mathsf{pk} = (g, h, u, v, g^{\alpha}, \ldots, g^{\alpha^n}, h^{\alpha}, \ldots, h^{\alpha^n}, u^{\alpha}, \ldots, u^{\alpha^n}, v^{\alpha}, \ldots, v^{\alpha^n}, f(x))$. Note that $\mathcal{B}$ can compute $u^{\alpha^i} = g^{\beta_0\alpha^i} \prod\limits_{j=0}^{n-1} (g^{\alpha^{i+j+1}})^{\beta_{0,j}} = g^{\sum\limits_{j=0}^{n-1} \beta_{0,j}\alpha^{i+j+1}} \cdot g^{\beta_0\alpha^i} = g^{\alpha^i Q(\alpha)}$. As $P(x), \gamma, \beta_0$ were chosen randomly so the simulated pk and and the one in the original scheme follows the same distribution. Hence the simulation is perfect.
Query phase 1: $\mathcal{A}$ adaptively issues key generation queries for the ID's of his choice. Simulate the secret key sk_i for ID_i as follows.

- Set $C_i(x) = \sum\limits_{j=0}^{n-2} \beta_{i,j} x^j$ and $D_i(x) = x(x - ID_i)C_i(x) + \beta_i$ where $\beta_{i,j}, \beta_i \xleftarrow{\$} \mathbb{Z}_p^*$.
- Compute $\mathsf{sk}_{ID_i} = (\mathsf{sk}_{i,1}, \mathsf{sk}_{i,2}, \mathsf{sk}_{i,3}, \mathsf{sk}_{i,4})$ as,

$$\mathsf{sk}_{i,1} = \left(h^{C_i(\alpha)}\right)^{\frac{1}{\gamma}} \qquad \mathsf{sk}_{i,2} = -D_i(ID_i) = -\beta_i$$

$$\mathsf{sk}_{1,3} = h^{Q(\alpha)} \qquad \mathsf{sk}_{i,5} = \left(w_i = h^{D_i(\alpha)}, w_i^{\alpha} = h^{\alpha D_i(\alpha)}, \ldots, w_i^{\alpha^n} = h^{\alpha^n D_i(\alpha)}\right).$$

$$\mathsf{sk}_{i,4} = \left(h^{\frac{\beta_i}{\beta_0} P(\alpha) + (\alpha - ID_i)C_i(\alpha)}\right)^{\frac{1}{\gamma}}$$

Two things are left to show that adversary can compute sk_i in this way and the computed sk_i is a valid sk_i. The first condition can be verified as follows,

$$\mathsf{sk}_{i,1} = \left(\prod_{j=0}^{n-2} h^{\beta_{i,j}\alpha^j}\right)^{\frac{1}{\gamma}} = \left(h^{\sum\limits_{j=0}^{n-2} \beta_{i,j}\alpha^j}\right)^{\frac{1}{\gamma}} = \left(h^{C_i(\alpha)}\right)^{\frac{1}{\gamma}}$$

$$\mathsf{sk}_{i,2} = -ID_i(ID_i - ID_i)C_i(x) - \beta_i = \beta_i$$

$$\mathsf{sk}_{i,3} = h^{\beta_0} \prod_{j=0}^{n-1} (h^{\alpha^{j+1}})^{\beta_{0,j}} = h^{\sum\limits_{j=0}^{n-1} \beta_{0,j}\alpha^{j+1} + \beta_0} = h^{\alpha P(\alpha) + \beta_0} = h^{Q(\alpha)} = z$$

$$\mathsf{sk}_{i,4} = \left(\prod_{j=0}^{n-1} h^{\frac{\beta_i}{\beta_0}\beta_{0,j}\alpha^j} \cdot \prod_{j=0}^{n-2} h^{\beta_{i,j}(\alpha - ID_i)\alpha^j}\right)^{\frac{1}{\gamma}} = \left(h^{\frac{\beta_i}{\beta_0} P(\alpha) + (\alpha + ID_i)C_i(\alpha)}\right)^{\frac{1}{\gamma}}$$

$$w_i^{\alpha^k} = h^{\beta_i\alpha^k} \prod_{j=0}^{n-2} \left(h^{\alpha^{j+k+2}}\right)^{\beta_{i,j}} \left(h^{\alpha^{j+k+1}}\right)^{ID_i\beta_{i,j}} = h^{\alpha^k D_i(\alpha)}, \text{ where } i \in [n] \text{ and } 0 \leq k \leq n.$$

It is easy to see that $\mathcal{B}$ can compute this using the values provided in the problem instance. The only thing left to show is that the simulation is correct. This can be verified as follows,

$$\begin{gathered}
\mathsf{sk}_{i,1} = \left(h^{C_i(\alpha)}\right)^{\frac{1}{\gamma}} = h^{\frac{D_i(\alpha)-\beta_i}{\alpha\gamma(\alpha-ID_i)}} = h^{\frac{D_i(\alpha)-D_i(ID_i)}{\alpha\gamma(\alpha-ID_i)}} = \left(w_i h^{-D_i(ID_i)}\right)^{\frac{1}{\alpha\gamma(\alpha-ID_i)}} \\
\mathsf{sk}_{i,2} = -D_i(ID_i) = -ID_i(ID_i - ID_i)C_i(x) - \beta_i = -\beta_i \\
\mathsf{sk}_{i,3} = h^{Q(\alpha)} \\
\mathsf{sk}_{i,4} = \left(h^{\frac{\beta_i}{\beta_0}(Q(\alpha)-\beta_0)+D_i(\alpha)-\beta_i}\right)^{\frac{1}{\alpha\gamma}} = h^{\frac{\frac{\beta_i}{\beta_0}Q(\alpha)+D_i(\alpha)}{\alpha\gamma}} = \left(z^{f(\mathsf{sk}_{i,2})}w_i\right)^{\frac{1}{\alpha\gamma}} \\
\mathsf{sk}_{i,5} = \{w_i = h^{D_i(\alpha)}, w_i^{\alpha} = h^{\alpha D_i(\alpha)}, \ldots, w_i^{\alpha^n} = h^{\alpha^n D_i(\alpha)}\}.
\end{gathered}$$

Now the distribution of sk_i also follows the distribution of the original construction as $P(x), \beta_0, C_i(x), \beta_i, \gamma$ for $i \in [n]$ are randomly chosen.

Challenge: $\mathcal{A}$ sends (S^*, M_0, M_1) to $\mathcal{B}$ where the identities of S^* were never queried in phase 1. $\mathcal{B}$ choses $b \xleftarrow{\$} \{0,1\}$ and sets,

$$\mathsf{ct}'_1 = Z^{-1} \qquad \mathsf{ct}'_2 = (g')^{-\alpha^{q+2}}$$

$$\mathsf{ct}'_3 = M_b \cdot Z^{\beta_0} \cdot e(g^{P(\alpha)}, h'^{\alpha^{q+2}}) \qquad \mathsf{ct}'_4 = \left(g'^{\alpha^{q+2}}\right)^{\gamma \prod_{i \in S^*}(\alpha - ID_i)\cdot}$$

It is easy to verify that $\mathcal{B}$ can simulate this from the problem instance it got. We need to verify that the simulation is correct. Let $s' = \log_g g' \cdot \alpha^{q+1}$ now if $Z = e(u,h)^{\alpha^{q+1}}$ then,

$$\begin{gathered}
\mathsf{ct}'_1 = e(g', h)^{-\alpha^{q+1}} = e(g,h)^{-s'}; \quad \mathsf{ct}'_2 = (g')^{-\alpha^{q+2}} \\
\mathsf{ct}'_3 = M_b \cdot e(g^{Q(\alpha)}, h'^{\alpha^{q+1}}) = M_b \cdot e(u,h)^{s'} \\
\mathsf{ct}'_4 = \left(g^{s'\alpha}\right)^{\gamma \prod_{i \in S^*}(\alpha - ID_i)} = \left(v^{s'}\right)^{\prod_{i \in S^*}(\alpha - ID_i)}.
\end{gathered}$$

Now as $\log_g g', \alpha$ were randomly chosen so s' is also random and follows the same distribution as the original scheme.

Query phase 2: $\mathcal{A}$ adaptively issues secret key query of $ID_i \notin S^*$ and $\mathcal{B}$ simulate and send sk_i same as query phase 1.

Guess: $\mathcal{A}$ stops and outputs $b' \in \{0,1\}$. If $b' = b$, $\mathcal{B}$ outputs 1 indicating $Z = e(g', h)^{\alpha^{q+1}}$ else outputs 0. Now, the advantage of $\mathcal{B}$ in the $\mathsf{waABDHE}$ game (we denote it by $\Pr\left[\mathcal{B}^{\mathsf{waABDHE}} \Rightarrow 1\right]$, which means $\mathcal{B}$ given the $\mathsf{waABDHE}$ instance outputs 1) is same as the adversary $\mathcal{A}$ guessing b with probability anything other than guessing randomly. So,

$$
\begin{aligned}
\Pr\left[\mathcal{B}^{\mathsf{waABDHE}} \Rightarrow 1\right] &\geq \Pr\left[\mathcal{A}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{bed}_{\mathcal{P}}} \Rightarrow 1\right] \\
\Pr\left[\mathcal{B}^{\mathsf{waABDHE}} \Rightarrow 1\right] &\geq \Pr\left[\mathcal{A}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{bed}_{\mathcal{P}}} \Rightarrow 1 | Z = g^{ab}\right] \Pr\left[Z = g^{ab}\right] \\
&\quad + \sum_{p-1} \Pr\left[\mathcal{A}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{bed}_{\mathcal{P}}} \Rightarrow 1 | Z \overset{\$}{\leftarrow} \mathbb{G}\right] \Pr\left[Z \overset{\$}{\leftarrow} \mathbb{G}\right] \\
\frac{1}{p}\Pr\left[\mathcal{A}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{bed}_{\mathcal{P}}} \Rightarrow 1 | Z = g^{ab}\right] &\leq \left|\Pr\left[\mathcal{B}^{\mathsf{waABDHE}} \Rightarrow 1\right] - \frac{p-1}{p}\Pr\left[\mathcal{A}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{bed}_{\mathcal{P}}} \Rightarrow 1 | Z \overset{\$}{\leftarrow} \mathbb{G}\right]\right| \\
\frac{1}{p}\Pr\left[\mathcal{A}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{bed}_{\mathcal{P}}} \Rightarrow 1 | Z = g^{ab}\right] &\leq \left|\Pr\left[\mathcal{B}^{\mathsf{waABDHE}} \Rightarrow 1\right] - \frac{p-1}{2p}\right| \\
\frac{1}{p}\Pr\left[\mathcal{A}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{bed}_{\mathcal{P}}} \Rightarrow 1 | Z = g^{ab}\right] &\leq \left|\Pr\left[\mathcal{B}^{\mathsf{waABDHE}} \Rightarrow 1 | Z = g^{ab}\right] \Pr\left[Z = g^{ab}\right]\right. \\
&\quad \left. + \sum_{p-1} \Pr\left[\mathcal{B}^{\mathsf{waABDHE}} \Rightarrow 1 | Z \overset{\$}{\leftarrow} \mathbb{G}\right] \Pr\left[Z \overset{\$}{\leftarrow} \mathbb{G}\right] - \frac{p-1}{2p}\right| \\
\Pr\left[\mathcal{A}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{bed}_{\mathcal{P}}} \Rightarrow 1 | Z = g^{ab}\right] &\leq \left|\Pr\left[\mathcal{B}^{\mathsf{waABDHE}} \Rightarrow 1 | Z = g^{ab}\right] + (p-1)\left(\frac{1}{2} - \frac{1}{2}\right)\right| \\
\mathsf{Adv}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathcal{A},\mathsf{bed}_{\mathcal{P}}} &\leq \mathsf{Adv}^{\mathsf{waABDHE}}_{\mathcal{B}}
\end{aligned}
$$

□

8 Conclusion

In this paper, we have shown the limitations of the existing works. Precisely, we found security issues in all of them. We formalized the definition of broadcast encryption with dealership and introduced a security requirement necessary in the real world. We propose three new constructions that achieve security guarantees required from a broadcast encryption with dealership to be deployed in real life. The three constructions, in a way, suggest a trade-off in terms of parameter size, efficiency and security. In this work, we only have considered *semi-honest but curios* adversaries without any collusion across entities. For possible future work, we suggest removing these restrictions without hampering the efficiency. Moreover, due to the highly interactive nature of BED, we could only achieve some of the proofs in the generic group model. We put forward the suggestion of getting standard assumption-based proof as another possible future work.

References

AB09. Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, New York, NY, USA, 1st edition, 2009.

AD16. Kamalesh Acharya and Ratna Dutta. Secure and efficient construction of broadcast encryption with dealership. In *Provable Security*, volume 10005 of *LNCS*, pages 277–295. Springer, 2016.

AD17. Kamalesh Acharya and Ratna Dutta. Adaptively secure broadcast encryption with dealership. In *Information Security and Cryptology*, volume 10157 of *LNCS*, pages 161–177. Springer, 2017.

BBW06. Adam Barth, Dan Boneh, and Brent Waters. Privacy in encrypted content distribution using private broadcast encryption. In *International Conference on Financial Cryptography and Data Security*, pages 52–64. Springer, 2006.

BGW05. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, volume 3621 of *LNCS*, pages 258–275. Springer, 2005.

Cam13. Philippe Camacho. Fair exchange of short signatures without trusted third party. In *Cryptographers' Track at the RSA Conference*, pages 34–49. Springer, 2013.

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.

CM14. Melissa Chase and Sarah Meiklejohn. Déjà Q: Using dual systems to revisit q-type assumptions. In *EUROCRYPT*, volume 8441 of *LNCS*, pages 622–639. Springer, 2014.

CM19. Sanjit Chatterjee and Sayantan Mukherjee. Large universe subset predicate encryption based on static assumption (without random oracle). In *CT-RSA*, volume 11405 of *LNCS*, pages 62–84. Springer, 2019.

Del07. Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In *ASIACRYPT*, volume 4833 of *LNCS*, pages 200–215. Springer, 2007.

Duc10. Léo Ducas. Anonymity from asymmetry: New constructions for anonymous HIBE. In Josef Pieprzyk, editor, *Cryptographers' Track at the RSA Conference*, volume 5985 of *LNCS*, pages 148–164. Springer, 2010.

FHS15. Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In *CRYPTO*, volume 9216 of *LNCS*, pages 233–253. Springer, 2015.

Fis99. Marc Fischlin. Pseudorandom function tribe ensembles based on one-way permutations: Improvements and applications. In *EUROCRYPT*, volume 1592 of *LNCS*, pages 432–445. Springer, 1999.

FN94. Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO*, volume 773 of *LNCS*, pages 480–491. Springer, 1994.

GLR18. Junqing Gong, Benoît Libert, and Somindu C Ramanna. Compact IBBE and Fuzzy IBE from Simple Assumptions. In *SCN*, volume 11035 of *LNCS*, pages 563–582. Springer, 2018.

GMSV13. Fuchun Guo, Yi Mu, Willy Susilo, and Vijay Varadharajan. Membership encryption and its applications. In *Australasian Conference on Information Security and Privacy*, pages 219–234. Springer, 2013.

GSP$^+$16. Clémentine Gritti, Willy Susilo, Thomas Plantard, Kaitai Liang, and Duncan S. Wong. Broadcast encryption with dealership. *International Journal of Information Security*, 15(3):271–283, 2016.

GW09. Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 171–188. Springer, 2009.

HILL99. J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

KLEL18. Joon Sik Kim, Youngkyung Lee, Jieun Eom, and Dong Hoon Lee. Recipient revocable broadcast encryption with dealership. In *Information Security and Cryptology*, volume 10779 of *LNCS*, pages 214–228. Springer, 2018.

LG18. Jiangtao Li and Junqing Gong. Improved anonymous broadcast encryptions. In *ACNS*, volume 10892 of *LNCS*, pages 497–515. Springer, 2018.

LPQ12. Benoît Libert, Kenneth G Paterson, and Elizabeth A Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In *International Workshop on Public Key Cryptography*, pages 206–224. Springer, 2012.

RS16. Somindu C. Ramanna and Palash Sarkar. Efficient adaptively secure IBBE from the SXDH assumption. *IEEE Transactions on Information Theory*, 62(10):5709–5726, 2016.

RWZ12. Yanli Ren, Shuozhong Wang, and Xinpeng Zhang. Non-interactive dynamic identity-based broadcast encryption without random oracles. In *Information and Communications Security*, volume 7618 of *LNCS*, pages 479–487. Springer, 2012.

Sch80. Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 256–266. Springer, 1997.

Wee16. Hoeteck Wee. Déjà Q: Encore! un petit IBE. In *TCC-II*, volume 9563 of *LNCS*, pages 237–258. Springer, 2016.

A Weaker asymmetric augmented bilinear Diffie-Hellman assumption

Weaker augmented bilinear Diffie-Hellman problem was introduced by [RWZ12]. It says, let $\mathcal{BG} = (p, g, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{\$} \mathsf{BGen}$ be a symmetric bilinear pairing where $\mathbb{G}$ and $\mathbb{G}_T$ is cyclic group of prime order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a Type-1 pairing. Then the wABDHE assumption is the following.

Definition 7. *Given* $\left(p, g, g^{\alpha}, \ldots, g^{\alpha^{n}}, g', (g')^{\alpha^{q+2}}, \ldots, (g')^{\alpha^{2q}}, Z\right) \in \mathbb{G}^{2q+1} \times \mathbb{G}_T$ *decide* $Z = e(g, g')^{\alpha^{q+1}}$ *or a random element from* $\mathbb{G}_T$*, where* g *an* g' *are two random generators of* $\mathbb{G}$.

For any adversary $\mathcal{A}$ the advantage of solving wABDHE is the following,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{wABDHE}}_{\mathcal{A},\mathsf{BGen}} := \Big| &\Pr\Big[\mathcal{A}\Big(g, g^{\alpha}, \ldots, g^{\alpha^n}, g', (g')^{\alpha^{q+2}}, \ldots, (g')^{\alpha^{2q}}, e(g, g')^{\alpha^{q+1}} \Rightarrow 1\Big)\Big] \\ &- \Pr\Big[\mathcal{A}\Big(g, g^{\alpha}, \ldots, g^{\alpha^n}, g', (g')^{\alpha^{q+2}}, \ldots, (g')^{\alpha^{2q}}, Y \in \mathbb{G}_T\Big) \Rightarrow 1\Big]\Big| \leq \epsilon\end{aligned}$$

To justify the assumption, we give a formal proof in Theorem 12. The proof requires us to use a lemma from [CM11]. We state the said lemma in Lemma 4 for completeness and omit the proof here. For a formal proof of Lemma 4, the reader is requested to check [CM11].

Lemma 4. *Let g, h', h be generators of $\mathbb{G}$, $\mathbb{H}'$ and $\mathbb{H}$ where $g = \psi(h')$ and $h = (\rho(h'))^{1/c}$ for some $c \in \mathbb{Z}_p^*$, then $e'(g, h') = \hat{e}(g, h)^{2c}$, where ψ and ρ are two efficiently computable isomorphism with $\rho(h') = h^c$.*

Theorem 12. *Let there exists a* ppt *adversary $\mathcal{A}$ that solves* $\mathsf{waABDHE}$ *problem in $\mathcal{PG} = (p, g, h, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) \xleftarrow{\$}$* PBGen *with non-negligible advantage then there exists an* ppt *algorithm $\mathcal{B}$ that can break* wABDHE *problem in $\mathcal{BG} = (p, g, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{\$}$* BGen *with non-negligible advantage.*

Proof. An wABDHE adversary $\mathcal{B}$ would try to use $\mathsf{waABDHE}$ adversary $\mathcal{A}$ as subroutine. Given a problem instance $\Big(g, g^{\alpha}, \ldots, g^{\alpha^n}, g', (g')^{\alpha^{q+2}}, \ldots, (g')^{\alpha^{2q}}, Z\Big) \in \mathbb{G}^{2q+1} \times \mathbb{G}_T$, $\mathcal{B}$ simulate $\mathcal{A}$ as follows. $\mathcal{B}$ finds a group $\mathbb{H}'$ isomorphic to $\mathbb{G}$, where $\psi : \mathbb{G} \to \mathbb{H}'$ is the isomorphism. $\mathcal{B}$ can find such isomorphism is polynomial time. Let $h' = \psi(g')$. Now re-arranging the problem instance we get,

$$\Big(g, g^{\alpha}, \ldots, g^{\alpha^n}, h', (h')^{\alpha^{q+2}}, \ldots, (h')^{\alpha^{2q}}, Z\Big) \in \mathbb{G}^{q+1} \times (\mathbb{H}')^q \times \mathbb{G}_T.$$

The map e induces $e' : \mathbb{G} \times \mathbb{H}' \to \mathbb{G}_T$ in a natural way, $e'(g, h') = e(g, \psi^{-1}(h')) = g_t$, where $h' \in \mathbb{H}'$, $g_t \in \mathbb{G}_T$. The isomorphism ψ forces e' to be onto. A trivial consequence of defining e' this way makes e' bilinear, non-degenerate and computable. Now given $\mathcal{A}$ $\Big(g, g^{\alpha}, \ldots, g^{\alpha^n}, h', (h')^{\alpha^{q+2}}, \ldots, (h')^{\alpha^{2q}}, Z\Big) \in \mathbb{G}^{q+1} \times (\mathbb{H}')^q \times \mathbb{G}_T$, $\mathcal{B}$ outputs the same as $\mathcal{A}$ outputs.

Now the problem instance for $\mathcal{A}$, e' is a Type-2 pairing. Only thing left to show that we can convert a Type-2 instance to a Type-3 instance in a natural way. This part follows trivially from Lemma 4. Chatterjee and Menzes in their paper [CM11] argued that any cryptographic protocol describe using a Type-2 pairing and corrosponding hard problem instance can naturally be converted in to Type-3 pairing and the hardness assumption in Type-3 is equivalent to the one used in Type-2. □