IACR Communications in Cryptology
ISSN 3006-5496, Vol. 1, No. 3, 33 pages.

https://doi.org/10.62056/ak5txl86bm

Reinventing BrED: A Practical Construction

Formal Treatment of Broadcast Encryption with Dealership

Avishek Majumder[1] and Sayantan Mukherjee[2]

[1] UPES, Dehradun, India
[2] IIT Jammu, India

Abstract. Broadcast Encryption (BE) allows a sender to send an encrypted message to multiple receivers. In a typical broadcast encryption scenario, the broadcaster decides the set of users who can decrypt a particular ciphertext (denoted as the privileged set). Gritti et al. (IJIS'16) introduced a new primitive called *Broadcast Encryption with Dealership* (BrED), where the dealer decides the privileged set. A BrED scheme allows a dealer to buy content from the broadcaster and sell it to users. It provides better flexibility in managing a large user base. To date, quite a few different constructions of BrED schemes have been proposed by the research community.

We find that all existing BrED schemes are insecure under the existing security definitions. We demonstrate a concrete attack on all the existing schemes in the purview of the existing security definition. We also find that the security definitions proposed in the state-of-the-art BrED schemes do not capture the real world. We argue about the inadequacy of existing definitions and propose a new security definition that models the real world more closely. Finally, we propose a new BrED construction and prove it to be secure in our newly proposed security model.

Keywords: Broadcast Encryption with Dealership · Dealer · Broadcast Encryption · Pairing-based Cryptography

1 Introduction

Public key encryption (PKE) allows secure communication between two parties without any shared secret key. In the PKE setting, the receiver Alice publishes her *public key*, which the sender Bob uses to encrypt his message. The security of PKE ensures that none but Alice decrypts the message. Consider a scenario where Bob wants to send a message to a set of users. Broadcast Encryption (BE) [FN93, BGW05] addresses this problem of efficiently sending an encrypted message to a set of users (S), called the *privileged set*. Correctness of BE ensures any user from S can decrypt the ciphertext. However, the security ensures that any user outside S cannot decrypt the message even in collaboration with other *unprivileged users*.

Consider a related scenario where an organization like PayTV wants to broadcast digital content to a large group of people. It is challenging for a single organization to maintain such a large user base while adding reach to new users. Thus, we often see scenarios where a broadband distributor sells OTT subscriptions. A convenient solution is that the central organization outsource the role of managing subscriptions and adding new users to some sub-distributors (we call them *dealer* in our work). Users register with a dealer and purchase their subscription. The dealer then buys digital content rights in bulk from the broadcaster. The dealer provides the broadcaster with a token (referred to as *group*

E-mail: avishek.majumder1991@gmail.com (Avishek Majumder), csayantan.mukherjee@gmail.com (Sayantan Mukherjee)

Received: 2024-07-09 Accepted: 2024-09-02

token) for the set of subscribed users. With the group token, the broadcaster encrypts the content for the privileged set and broadcasts. This variant of BE was introduced by Gritti *et al.* [GSP+16] and was named *Broadcast Encryption with Dealership* (BrED).

There are three major entities in a typical BrED scheme: *broadcaster*, *dealers*, and *end-users*. Users who want digital content from a particular broadcaster register themselves with the respective dealers. Among the registered users, those who have purchased a subscription for some specific content are called *privileged users* (members of the set S). The dealer buys rights of digital content for the privileged user set S by providing a token for the set S. The dealer does not disclose the set S of users to the broadcaster for securing his business. The broadcaster encrypts and broadcasts its digital content for users in S, using the token provided by the dealer. The security of BrED should ensure that the broadcaster can verify the size of the set S without getting any other information about the set S.

In addition to the fundamental security prerequisites of BE schemes, which restrict the decryption of encrypted broadcasts to privileged users only, BrED introduces several new security considerations not present in conventional BE. Below, we informally outline and justify all the security requirements for BrED.

1. The primary security requirement of BrED is that the unprivileged users cannot decrypt the broadcasted ciphertext. This is also a requirement of BE, and in this work, we call it *message indistinguishability from unprivileged users*.

2. Additionally, in BrED, the dealer chooses the set S of privileged users, and the broadcaster creates ciphertext which can only be decrypted by users in S. However, the dealer does not reveal the set S to the broadcaster. As for its business motive, the dealer does not want the broadcaster to have access to its user base. Thus, the dealer transmits the information of S to the broadcaster through a quantity called the *group token*.

   It is required that the group token keeps the set S hidden to the broadcaster, still enabling the broadcaster to construct ciphertexts that can be decrypted by privileged users in S and be secure in terms of *message indistinguishability from unprivileged users*. The requirement that the group token does not reveal any information (except the size) about the set S to the broadcaster is called *group privacy*.

3. As the set S is kept hidden from the broadcaster, the broadcaster is required to know the cardinality of S. Without this knowledge, the dealer may make the broadcaster serve more users than it is paying for. Thus, given the group token, the broadcaster should be able to verify the cardinality of the set S which the token represents. This requirement is called *maximum user of accountability*.

The above security requirements were specified in [GSP+16], later rectified by [AD16], and followed thereafter. In Section 3, we show that the security definitions proposed for message indistinguishability from unprivileged users in both works do not model the practical world conclusively. In fact, there are some gaps in the security definition for message indistinguishability from unprivileged users proposed thus far, as none of them considered the presence of the dealer with sufficient formalization. In this paper, we introduce a new definition which bridges the gap. We discuss these gaps and new security definitions in Section 3.

1.1 Related Works

To our knowledge, the only works explicitly constructing broadcast encryption with dealership are [GSP+16, AD16, AD17, KLEL17, AD21]. Gritti *et al.* [GSP+16] was the first to introduce the notion of BrED and provide security definitions. In [GSP+16], a construction of BrED was also provided along with its security proof. It was claimed that the group privacy of the scheme was unconditionally secure. The message indistinguishability from unprivileged users was achieved in the semi-static setting. However, they claim that adaptive security can be achieved using a generic technique introduced in [GW09].

[AD16] proposed some improvements over [GSP+16]. In particular, in [AD16], the unconditional security claim of group privacy in [GSP+16] was challenged. The new construction of [AD16] claimed to achieve group privacy against a computationally bounded adversary. The scheme claimed to achieve message indistinguishability from users in the selective security model.

In [AD17], some improvements over [AD16] were proposed. In particular, message indistinguishability from the user was argued in the adaptive model. Kim *et al.* [KLEL17] modified the construction of [AD17] to support recipient revocation without compromising security. In [AD21], two more constructions were proposed, wherein one of them, the message indistinguishability from unprivileged users, was proved in the adaptive model, and the other claimed to achieve message indistinguishability from unprivileged users in the selective setting. All the above constructions claim to give cryptographic proof for maximum user of accountability. However, there is a flaw in the security argument, which we discuss in Section 3.2.

1.2 Our Contributions

In this paper, we do a comprehensive study of BrED and our contributions to this paper are multifold. We report concrete attacks (in Sec 3.1) on all previous constructions in their proposed security model. We also identify some gaps in the security model proposed in earlier works [GLR18, AD16]. We fill this gap by introducing a new security model and following it up with a secure construction under the new security definition (in Sec 4). Below we discuss our contributions in detail.

Attacks on previous constructions. In Section 3.1, we present concrete attacks on the *group privacy* of the existing schemes [AD16, AD17, KLEL17, AD21] in the purview of the existing security model. All the previous works claimed that their constructions achieve the *group privacy* security under the standard *discrete-log* assumption. However, we mount a pairing-based attack, rendering all those schemes insecure in their claimed security model. On top of this, we also identify critical errors in the *maximum user of accountability* security proof of all the existing schemes which we discuss in Section 3.2.

Formalizing definitions and introduction of new security notions. Secondly, the previous works on BrED did not consider the presence of a dealer with sufficient formalization. The dealer plays a significant role in BrED by selecting the privileged set and generating the group token. The broadcaster uses the group token in the encryption phase. Thus, the view of an unprivileged user and a dealer is quite different. All the previous works [GSP+16, AD16, AD17, KLEL17, AD21] have omitted that the dealer generates the group token in their *message indistinguishability from unprivileged users* security game which we argue to be a flaw.

This paper first puts forward the security definition of message indistinguishability from unprivileged users of BrED, acknowledging a dealer's presence with necessary formalization (see Sec 3). We propose a security model that captures an adversarial dealer and an unprivileged user in collaboration. The adversary in our security game $\mathsf{cpa}_{\mathcal{U}}$ thus considers both an unprivileged user and the dealer in collusion. Therefore, the definition we provide in this paper generalizes the security requirements of BrED that were missing in earlier works [GSP+16, AD16, AD17, KLEL17, AD21] towards capturing real-world requirements.

In this work, we consider a semi-honest dealer for maximum user of accountability, implying that the dealer adheres to the protocol only during group token generation but may attempt to issue a group token for a larger group size than agreed upon and also try to learn about the message from the ciphertext. In such cases, the broadcaster requires a way to verify the size of the group token. The security definition of the maximum user of accountability in [AD16, AD17] considers a semi-honest dealer, whereas [GSP+16, KLEL17, AD21] considered a malicious dealer. Even if some of the above works gave a maximum users of accountability security definition against malicious adversaries, we emphasize that all the existing works [GSP+16, AD16, AD17, KLEL17, AD21] proposed proofs of maximum users of accountability against semi-honest adversaries. Then again, we point out a common flaw in their security model as well as in their security arguments. We fix the model and prove maximum users of accountability of our construction in a generic group model.

A semi-honest adversary is weak but practical as the the broadband distributor (i.e. the dealer) often is not a cryptographer who can maliciously produce valid ciphertexts, but they may want to squeeze in a few users in the privileged set without increasing their committed cardinality to make a better profit. A more robust form of maximum user of accountability would address malicious dealers who may intentionally deviate from the protocol during group token generation. We leave this as an open problem.

Proposing a new construction. We then go on to propose the first secure construction of BrED. We provide the first secure prime-order construction in the proposed security model, achieving *constant-size ciphertext*. The security of our construction was achieved in the generic group model. However, we manage to achieve *group privacy* (priv) of our construction under the standard DDH assumption. Our construction of BrED is based on the identity-based broadcast encryption of Gong *et al.* proposed in [GLR18]. We believe this construction is a bit involved, and we give a brief technical overview of our construction in Section 4.

To summarize, we provide a concise list of our contributions next.

1. Firstly, we show concrete attacks on all the existing schemes [AD16, AD17, KLEL17, AD21], rendering all of them insecure in their proposed *group privacy* security game.
2. Then, we further show that the security proof for *maximum user of accountability* is incorrect in all the previous constructions.
3. Then, we formalize the existing definition and bridge the gap in the security requirements of message indistinguishability from unprivileged users.
4. We finally propose a construction of BrED. Our construction uses *prime-order bilinear pairing groups*. We prove our scheme to be secure in the newly introduced security definition.

1.3 Organization of Paper

In Section 2, we present the definitions and mathematical preliminaries. Next, in Section 3, we present an exposition of the vulnerabilities in the existing security definition of BrED and propose our modified notion of security. In the same section, we also show a concrete pairing-based attack on one of the previous works on BrED. The same attack holds on all the previous schemes. This is then followed by Section 4 where we propose our construction of BrED with constant-size ciphertext in a prime-order pairing group. Then, we conclude this paper in Section 5. To make our presentation self-contained, we have a well-marked appendix that contains some additional expositions. In particular, in Appendix A we revisit the description of previous schemes. Appendix B discusses the flaws we could find in the previous works. We have deferred this section to the appendix only to avoid repetition of the flaws, as several of the previous works suffer from similar issues.

2 Definitions and Preliminaries

We discuss some necessary tools required by the constructions in this paper.

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. A function $\mathsf{neg} : \mathbb{N} \to \mathbb{R}^+$ is called a negligible function if for all positive polynomial $q(\cdot)$, there exists $x_0 \in \mathbb{N}$ such that for all $x > x_0$, $\mathsf{neg}(x) < 1/q(x)$.

2.2 Groups and Hardness Assumptions

This section discusses different types of elliptic curve groups and hardness assumptions that we will require in this work. We next consider a set of elliptic curve groups where the bilinear pairing function is efficiently evaluated.

Bilinear Pairing. Let $\mathbb{G}$, $\mathbb{H}$, $\mathbb{G}_T$ be three commutative multiplicative groups of the order of a large prime p. A map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ 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 use Type-3 pairing where $\mathbb{G}$ and $\mathbb{H}$ have no known isomorphism.

2.2.1 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 of large prime p, $\mathbb{G} = \langle g \rangle$, $\mathbb{H} = \langle h \rangle$ and $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ is an admissible, non-degenerate asymmetric Type-3 bilinear pairing. In this work, we make use of the following hardness assumptions.

Decisional Diffie-Hellman Assumption (DDH) [RCS12] .

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

$$\mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{A},\mathcal{PG}}(\lambda) = \left|\Pr\left[\mathcal{A}(\mathcal{X}, g^{ab}) = 1\right] - \Pr\left[\mathcal{A}(\mathcal{X}, g^{c}) = 1\right]\right| \leq \mathsf{neg}(\lambda),$$

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

2.2.2 DeMillo-Lipton-Schwartz-Zippel (DLSZ) Lemma [DL78, Zip79, Sch80].

If a polynomial $p(x_1, x_2, \ldots, x_m)$ over $F = GF(q)$ is nonzero and has total degree at most d, then

$$\Pr\left[p(a_1, a_2, \ldots, a_m) \neq 0\right] \geq 1 - \frac{d}{q},$$

where the probability is over all choices of $a_1, a_2, \ldots, a_m \in F$.

2.2.3 Generic Group Model.

The generic group model was explored formally first by Shoup [Sho97]. This technique proves the lower bounds of certain computational and decisional problems. This is explored regarding 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 Broadcast Encryption with Dealership

In the introduction section, we formally define a BrED scheme and then the role of each entity of a BrED system. The definition of BrED follows the definition provided in [GSP+16, AD16].

Definition 2. A BrED is a tuple of six ppt algorithms $\mathsf{BrED} = (\mathsf{BrED.Setup}, \mathsf{BrED.KeyGen}, \mathsf{BrED.GroupGen}, \mathsf{BrED.Verify}, \mathsf{BrED.Encrypt}, \mathsf{BrED.Decrypt})$.

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{BrED.Setup}(1^\lambda, n)$: It takes as input the maximal size n of the set of receivers per broadcast and the security parameter λ and outputs public parameter pp and a master secret key msk.

- $\mathsf{sk}_i \leftarrow \mathsf{BrED.KeyGen}(\mathsf{pp}, \mathsf{msk}, i)$: On invocation with $\mathsf{pp}, \mathsf{msk}$ and a user identity $i \in \mathbb{N}$, it outputs a user secret key sk_i for user i.

- $(\Gamma_S, k) \leftarrow \mathsf{BrED.GroupGen}(\mathsf{pp}, k, S)$: It takes as 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 (Γ_S, k) where Γ_S is a group token for the set S.

- $(0/1) \leftarrow \mathsf{BrED.Verify}(\mathsf{pp}, \Gamma_S, k)$: On input pp, a group token Γ_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.

$$\mathsf{BrED.Verify}(\mathsf{pp}, \Gamma_S, k) = \begin{cases} 1, & \text{if } |S| \leq k \\ 0, & \text{otherwise.} \end{cases}$$

- $\mathsf{CT}_S \leftarrow \mathsf{BrED.Encrypt}(\mathsf{pp}, \Gamma_S, M)$: It takes as input the public parameter pp and a group token Γ_S and a message M and outputs a ciphertext CT_S.

- $M \leftarrow \mathsf{BrED.Decrypt}(\mathsf{pp}, \mathsf{sk}_i, (S, \mathsf{CT}_S))$: On input pp, sk_i and a ciphertext CT_S for a set S, $\mathsf{BrED.Decrypt}$ outputs message M if $i \in S$.

Correctness. A BrED scheme is said to be correct if $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{BrED.Setup}(1^\lambda, n)$, for all $S \subseteq [n]$ such that $|S| \leq k \leq n$, $(\Gamma_S, k) \leftarrow \mathsf{BrED.GroupGen}(\mathsf{pp}, k, S)$, and $\mathsf{CT}_S \leftarrow \mathsf{BrED.Encrypt}(\mathsf{pp}, \Gamma_S, M)$ then for all $i \in S$, the following condition holds:

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

The major entities of BrED schemes are broadcasters, dealers, and end-users. One more entity named the *key generation centre* (KGC) is also required to construct a public-key BrED. Below, we elaborately discuss the role of each entity of a BrED scheme in light of the Definition 2.

1. **The KGC:** at the beginning, a KGC runs the $\mathsf{BrED.Setup}$ algorithm taking as input the security parameter 1^λ and the maximal size of the users per broadcast n, and produces $(\mathsf{pp}, \mathsf{msk})$. It publishes the public parameters pp and keeps the master secret key msk to itself. When invoked with a join request for a user i, the KGC computes the user's secret key sk_i using the $\mathsf{BrED.KeyGen}$ algorithms and sends the user's secret key to the user via a secure channel.
2. **The Dealer:** it selects a group of the user $S \subseteq [n]$, and an integer $k \leq n$ such that $|S| = k' \leq k$. It generates a group token Γ_S using the $\mathsf{BrED.GroupGen}$ algorithm and gives it to the broadcaster along with the value k. The dealer sends S to every user in the privileged set via a secure channel.
3. **The Broadcaster:** it verifies the group token with the $\mathsf{BrED.Verify}$ algorithm and upon verification generates the ciphertext CT_S using Γ_S for the set S and broadcasts CT_S.
4. **An End-User:** i-th user decrypts the ciphertext CT_S using their secret key sk_i if $i \in S$.

2.4 Security Definition of BrED

In the introduction section, we have mentioned the security requirements of a BrED system informally. This section recalls the formal definition of those security notions considered in the literature. We will mainly follow the definition given by [AD16]. In the next section (Section 3), we will discuss the flaw or incompleteness of the security requirements of [AD16].

2.4.1 Group Privacy.

In the BrED model, the dealer selects the privileged set and provides a group token for this set to the adversarial broadcaster. To maintain the dealer's business, the group token must secure the confidentiality of the members of a privileged set. Specifically, from the group token Γ_S, the broadcaster should not be able to infer any meaningful information about the underlying set $S \subseteq [n]$ other than its size. Group privacy security ensures that the privileged set remains secret from the adversary.

In the group privacy security game, the adversary (the broadcaster) is allowed to choose two sets of privileged users (S_0, S_1) of the same size. The experiment selects a random bit $\beta \in \{0, 1\}$, generates the group token for S_β, and sends it to the adversary. The adversary's task is to determine the random bit of the challenge. The formal security definition is as follows.

A BrED scheme BrED satisfies group privacy (priv) if for all ppt adversaries $\mathcal{A}$,

$$\mathrm{Adv}^{\mathsf{priv}}_{\mathcal{A},\mathsf{BrED}}(\lambda) = \left| \frac{1}{2} - \Pr\left[\mathsf{Exp}^{\mathsf{priv}}_{\mathsf{BrED}}(1^\lambda, \mathcal{A}) = 1\right] \right| \leq \mathsf{neg}(\lambda),$$

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

2.4.2 Maximum User of Accountability.

As previously discussed, for the dealer's business interests, the group privacy security of BrED requires that the adversarial broadcaster does not gain any information about

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

Figure 1: Group Privacy for BrED

the members of the privileged set from the group token, except the size of the privileged set. So, the broadcaster must ensure that the dealer has generated a group token for a set whose size does not exceed what the dealer has paid for. Thus, the maximum user of accountability security model to capture that a group token (Γ_{S^*}, k) cannot encode a privileged set S^* of size k^* bigger than a committed size k.

While defining MUA security, all the previous works on BrED formulated the adversary to submit the privileged set S^* along with the challenge group token Γ_{S^*}. The broadcaster behaves as the challenger in Maximum User of Accountability, submitting S^* is not inline with the security requirement of BrED. Looking ahead, we discuss this in Section 3.2.1. In Figure 2, we define MUA for a semi-honest dealer where the adversary is allowed to submit the group token Γ_{S^*}, and the commitment for the size of the set k only. A semi-honest dealer chooses any $k < n$ and a set $S^* \subseteq [n]$ of size strictly greater than k. The adversary then computes the group token Γ_{S^*} for S^* using $\mathsf{BrED.GroupGen}$ and to submit (Γ_{S^*}, k) as a challenge. The adversary wins only if the verification $\mathsf{BrED.Verify}$ outputs 1 for the parameters (Γ_{S^*}, k), and $\lvert S^* \rvert > k$.

A stronger maximum user of accountability security captures a malicious dealer that may deviate from the protocol for group token generation and submit an arbitrarily generated group token, which is not well-formed. Dealing with such an adversary necessitates that the challenger could verify that token Γ_{S^*} is really constructed for k users. In this work, we aim at a weaker security goal with efficiency where we aim to capture that a semi-honest dealer is not able to create a group token for a set S^* bigger than its committed k but still passes Verify.

Previous works on BrED [KLEL17, AD21] defined maximum user of accountability in the malicious adversarial model but could only prove it in the semi-honest adversarial model. We discuss those in detail in Section 3.2. Then again, the security arguments of previous works are not entirely correct, which we report in details in Section 3.2.2. We also mark achieving maximum user of accountability security in a malicious model as an open problem.

A BrED scheme BrED satisfies maximum user of accountability (mua) if for all ppt adversaries $\mathcal{A}$,

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

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

2.4.3 Message Indistinguishability for Unprivileged Users under CPA.

The final security requirement of BrED is that the unprivileged users can not distinguish between two ciphertexts of the same size, even in collusion with other unprivileged users. This security model captures that given a ciphertext CT_S, no unprivileged user can get any

Game Description
$\underline{\mathsf{Exp}^{\mathsf{mua}}_{\mathsf{BrED}}(1^\lambda, \mathcal{A})}$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{BrED.Setup}(1^\lambda, n)$
$(S^*, k) \leftarrow \mathcal{A}(\mathsf{pp})$, such that, $k < n$, $S^* \subseteq [n]$, and $\|S^*\| = k^* > k$
$\mathcal{A}$ computes $(\Gamma_{S^*}, k^*) \leftarrow \mathsf{BrED.GroupGen}(\mathsf{pp}, k^*, S^*)$
$\mathcal{A}$ submits (Γ_{S^*}, k) as a forgery
Return 1 if the following holds:
$\mathsf{BrED.Verify}(\mathsf{pp}, \Gamma_{S^*}, k) \to 1$

Figure 2: Maximum User of Accountability for BrED

Game Description	Oracle Description
$\underline{\mathsf{Exp}^{\mathsf{cpa}_\mathcal{X}}_{\mathsf{BrED}}(1^\lambda, \mathcal{A})}$	$\underline{\mathcal{O}_{\mathsf{sk}}(i)}$
$\mathcal{Q}_{\mathsf{sk}} \leftarrow \emptyset$	$\mathcal{Q}_{\mathsf{sk}} \leftarrow \mathcal{Q}_{\mathsf{sk}} \cup \{i\}$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{BrED.Setup}(1^\lambda, n)$	Run $\mathsf{sk}_i \leftarrow \mathsf{BrED.KeyGen}(\mathsf{msk}, i)$
$(S, M_0, M_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sk}}(\cdot)}(\mathsf{pp})$ such that $\|S\| \le k \le n$	Return sk_i
$\Gamma_S \leftarrow \mathsf{BrED.GroupGen}(\mathsf{pp}, k, S)$	
Sample $\beta \xleftarrow{\$} \{0,1\}$, $\mathsf{CT}_{S,\beta} \leftarrow \mathsf{BrED.Enc}(\mathsf{pp}, \Gamma_S, M_\beta)$	
$\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sk}}(\cdot)}(\mathsf{pp}, \mathsf{CT}_{S,\beta})$	
Return 1 if $(\beta = \beta')$ AND $(\mathcal{Q}_{\mathsf{sk}} \cap S = \emptyset)$.	

Figure 3: Adaptive Message Indistinguishability from Unprivileged Users for BrED

information about the underlying message M even if they collude. A BrED scheme BrED satisfies message indistinguishability for unprivileged user ($\mathsf{cpa}_\mathcal{X}$) if for all ppt adversaries $\mathcal{A}$,

$$\mathrm{Adv}^{\mathsf{cpa}_\mathcal{X}}_{\mathcal{A},\mathsf{BrED}}(\lambda) = \left| \frac{1}{2} - \Pr\left[\mathsf{Exp}^{\mathsf{cpa}_\mathcal{X}}_{\mathsf{BrED}}(1^\lambda, \mathcal{A}) = 1\right] \right| \le \mathsf{neg}(\lambda),$$

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

The above definition follows the definitions provided in [GSP+16, AD16]. The experiment defined in Figure 3 represents the adaptive $\mathsf{cpa}_\mathcal{X}$ security game. In Section 3.3, we demonstrate that this security game does not accurately reflect real-world scenarios, and we propose a modified security definition.

3 Inadequacy of Existing Schemes

In this section, we provide a concise critique of existing works. Firstly, we present an attack on group privacy in the existing model that makes all the existing constructions insecure. We also discuss a critical flaw in the security argument of maximum user of accountability of the existing works. Then, we take a look at the existing BrED security definitions and argue the necessity of a new security definition. Finally, we formalize the new security definition. To keep our presentation simple, we show the attack on [AD16] in this section and defer the attack on other works [AD17, KLEL17, AD21] to Appendix B.

3.1 Breaking BrED: Pairing-based Attack on Existing BrEDs

We observed that all existing schemes [AD16, AD17, KLEL17, AD21] argued group privacy similarly. In particular, all claimed group privacy under the standard discrete-log problem (DLP). However, in all four constructions, the group tokens are essentially in the span of the public key, where the span is determined by the privileged set information S. Since all four constructions are instantiated in the Type-1 pairing groups where DDH is easy, any efficient adversary could easily decide β in the group privacy security model (see Figure 1). Here, we show the exact attack for [AD16] and include the attack details of the rest of the works [AD17, KLEL17, AD21] in Appendix B for completeness.

3.1.1 Attack on group privacy of [AD16].

Let us recall the group privacy security model in Figure 1. Informally speaking, the adversary selects two sets S_0, S_1 of equal size k, to which the challenger chooses $\beta \in \{0, 1\}$ randomly and provides the adversary with group token Γ_{S_β}.

Let, $\mathcal{BG} = (p, \mathbb{G}, \mathbb{G}_T, e)$ be a bilinear group system, where $\mathbb{G}, \mathbb{G}_T$ are group of prime order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an admissible Type-1 bilinear pairing. $H : \{0, 1\}^* \to \mathbb{Z}_p^*$ is a cryptographically secure hash function. Authors of [AD16] define the public parameter pp and the group token Γ_{S_β} (w.r.t the challenge privileged user set S_β) as following:

$$\begin{aligned}
\mathsf{pp} &= \left(\mathcal{BG}, g, g^{\alpha}, \ldots, g^{\alpha^n}, e(g, h), h^{\alpha}, H, \mathsf{ID}\right), \\
\Gamma_{S_\beta} &= (\omega_1, \omega_2, \omega_3, \omega_4) \\
&= (h^{-s\alpha}, g_{n-k}^{sP_{S_\beta}(\alpha)}, g^{sP_{S_\beta}(\alpha)}, e(g, h)^s) \\
&= (h^{-s\alpha}, g^{s\alpha^{n-k} P_{S_\beta}(\alpha)}, g^{sP_{S_\beta}(\alpha)}, e(g, h)^s)
\end{aligned}$$

where g and h are random generators of $\mathbb{G}$, $P_S(x) = \prod_{ID_j \in S_\beta \subseteq [n]} (x + H(ID_j))$, and $s \xleftarrow{\$} \mathbb{Z}_p$ was sampled by the challenger.

Given pp and Γ_{S_β}, we mount the following attack:

1. Adversary knows both S_0 and S_1.
2. Thus, can compute $g^{P_{S_0}(\alpha)}$ from pp as $k \leq n$.
3. Adversary evaluates the following pairings,

$$\begin{aligned}
e\left(g^{P_{S_0}(\alpha)}, \omega_1\right) &= e\left(g, h\right)^{-s\alpha P_{S_0}(\alpha)} \\
e\left(\omega_3, h^{\alpha}\right) &= e(g, h)^{s\alpha P_{S_b}(\alpha)}.
\end{aligned}$$

4. Adversary finally decide

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

Note that, if the random bit β 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 can always decide β correctly.

Hence, the construction of [AD16] could not achieve group privacy because of the use of a Type-1 bilinear pairing group. However, inspired by the construction of [Duc10] we instantiate our construction in Section 4 in a Type-3 bilinear pairing group which helped us achieve group privacy.

3.2 Issue with Maximum User of Accountability

In this section, we revisit the issues with of the MUA security model and its corresponding proof as described in the existing literature. There are primarily two issues with the existing works: one concerning its security framework, and the other is with the proof technique. We discuss these issues one by one in this section.

3.2.1 Flaw in MUA Security Model

Recall that, S denotes the privileged set and k be the maximum allowed size for S. The maximum user of accountability security ensures that a group token Γ_S cannot pass the verification stage if $|S| > k$. Previous constructions on BrED, precisely [GSP+16] defined MUA security in malicious adversarial model. The authors of [AD16, AD17] defined MUA in semi-honest adversarial model. Again [KLEL17, AD21] defined and claimed to prove MUA in malicious model. However, all the works, the proof only captures a semi-honest adversary. All existing works the adversary generates the challenge group token in the MUA security proof following the actual protocol. Their security argument also had a flaw, which we describe next.

All the previous works on BrED the adversary submits (Γ_{S^*}, S^*) as a forgery, where S^* is the group chosen by the adversary. In MUA security game, the challenger is the broadcaster. Submitting the group S^* violates the basic security requirement of BrED. Therefore it is essential to not submit S^* as the challenge security game in Figure 2.

3.2.2 Flaw in MUA Security Proof

To argue MUA security, existing works [GSP+16, AD16, AD17, KLEL17, AD21] used the standard (f, n)-DHE assumption as defined in [GSP+16]. The (f, n)-DHE problem is the following.

> *The (f, n)-Diffie-Hellman Exponent Assumption*: given an instance $(\mathcal{G}, g, g^{\alpha}, \ldots, g^{\alpha^n})$ 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 constructions [GSP+16, AD17, KLEL17, AD21] also use a similar argument. To explain the problem in the proof of [AD16], we give a gist of their proof below.

The proof assumes an adversary $\mathcal{A}$, that breaks the *maximum user of accountability* of [AD16]. Let $\mathcal{B}$ be another adversary trying to solve the (f, n)-DHE problem. The 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. $\mathcal{A}$ computes a privileged set S^* such that $|S^*| = k' > k$ and generates a group token,

$$\begin{aligned}\Gamma_{S^*} &= (\omega_1, \omega_2, \omega_3, \omega_4) \\ &= \left(h^{-s\alpha}, g^{s\alpha^{n-k} P_{S^*}(\alpha)}, g^{sP_{S^*}(\alpha)}, e(g, h)^s\right),\end{aligned}$$

where $s \xleftarrow{\$} \mathbb{Z}_p$, $P_{S^*}(x) = \prod_{ID \in S^*} (x + H(ID))$. $\mathcal{A}$ sends (Γ_{S^*}, S^*) to $\mathcal{B}$. At this point, the adversary $\mathcal{B}$ sets $f(x) = sx^{n-k} P_{S^*}(x)$ and returns $\left(f(x), g^{s\alpha^{n-k} P_{S^*}(\alpha)}\right)$ as a solution to (f, N)-DHE problem. Note that $P_{S^*}(x)$ is a polynomial of degree k'. So, $f(x)$ is a polynomial of degree $n - k + k' > n$ as $k' > k$.

More precisely, the authors of [AD16] claimed that if the adversary $\mathcal{A}$ outputs a valid forgery (that is the group token passes the verification), then, $\big(f(x), \widehat{\omega}_2 = g^{f(\alpha)}\big)$ is the (f, N)-DHE solution for $\mathcal{B}$. So if the adversary $\mathcal{A}$ successfully breaks the maximum user of accountability of their scheme, then the adversary $\mathcal{B}$ breaks the (f, N)-DHE problem, which is believed to be "hard" respective to $\mathbb{G}$. For a detailed proof, readers are requested to go through [AD16].

One important thing to note here is that, the adversary submits a group token generated using the actual protocol indicating a semi-honest dealer. The adversary (the dealer) selects a challenge privileged set S^* and a size k such that $|S^*| > k$. It then samples a secret s from $\mathbb{Z}_p$. The adversary generates the group token Γ_{S^*} using BrED.GroupGen (indicating a semi-honest dealer) and the sampled secret s. Finally, the adversary submits (Γ_{S^*}, S^*) to the challenger which in this case is the (f, n)-DHE. Even through we allow submitting S^* without disclosing the secret s used to generate Γ_{S^*}, the (f, n)-DHE adversary $\mathcal{B}$ cannot compute

$$f(x) = sx^{n-k} P_{S^*}(x).$$

Hence, the proof does not hold. The MUA proof of [GSP+16] did submit the randomness s along with S^* and Γ_{S^*} to allow the adversary $\mathcal{B}$ construct a (f, n)-DHE solution, this violates their security game.

Finally, we conclude that, if the adversary $\mathcal{A}$ does not submit S^* (the privileged user set) and/or s (the randomness) with its forgery, then the security proof does not hold. As $\mathcal{B}$ cannot construct $P_{S^*}(x)$ as well as $f(x)$ without knowing S^* and/or s respectively. Hence, the simulation they provided is incorrect.

3.3 Inadequacy of $\mathsf{cpa}_{\mathcal{X}}$ Security Model

None of the previous works on BrED [GSP+16, AD16, AD17, KLEL17, AD21] considered the dealer as a potential adversary. In particular, the *message indistinguishability from unprivileged users under CPA* security (i.e. the $\mathsf{cpa}_{\mathcal{X}}$ security described in Figure 3) does not capture the fact that an adversarial dealer who generated the group token Γ_S (for a set $S \subseteq [n]$), could extract meaningful information about the plaintext M after seeing the ciphertext $\mathsf{CT}_S \leftarrow \mathsf{Enc}(\mathsf{pp}, \Gamma_S, M)$.

More precisely, in Figure 3, an adversary (with access to the secret key oracle) submits a challenge (S, M_0, M_1). The challenger generates group token Γ_S in response. The challenger then selects a bit $\beta \leftarrow \{0, 1\}$, and encrypts M_β with the help of Γ_S, and generates the challenge ciphertext $\mathsf{CT}_{S,\beta} \leftarrow \mathsf{BrED.Enc}(\mathsf{pp}, \Gamma_S, M_\beta)$. Finally, the challenger only provides the challenge ciphertext to the adversary and nothing else. This essentially models an honest dealer. Also, the group token is sent from the dealer to the broadcaster via a public channel. So everyone has access to that. The security game in Figure 3 does not capture this either, as we only provide the adversary with the ciphertext CT_S and not the group token Γ_S. This, in our opinion, does not correctly model a practical scenario as if a dealer can extract the underlying plaintext M (or any related information), it could easily sell it to more users without the broadcaster's knowledge.

3.4 $\mathsf{cpa}_{\mathcal{U}}$: A Stronger Security Model for BrED

It may seem that providing access to group tokens to the adversary of $\mathsf{cpa}_{\mathcal{X}}$ may fix the security definition. Unfortunately, this is not the case. A crucial point that was missed in all the previous works was that the dealer is never a part of the privileged set. This way, a dealer is also an unprivileged user. A more striking fact is that the dealer in a BrED system generates the group token and provides it to the broadcaster. The security requirements for a BrED scheme should ensure that the dealer who forms the group token does not infer any information about the message from the ciphertext.

Below, we formally introduce the modified security definition. Henceforth, we call this as *message indistinguishability from unauthorized users* ($\mathsf{cpa}_{\mathcal{U}}$) i.e. both the dealer and the unprivileged users. In the new security definition, upon seeing the public parameter, we allow the adversary to select a privileged set $S \subseteq [n]$ of size k and two messages M_0 and M_1 of equal size. The adversary with access to the secret key oracle then computes the group token Γ_S using $\mathsf{BrED.GroupGen}$ and provides (Γ_S, k, M_0, M_1) as its challenge. Here again, we consider a semi-honest adversary, as it honestly computes the group token. The challenger then selects a random bit $\beta \in \{0,1\}$ and provides the adversary with the ciphertext w.r.t. M_β. To which the adversary has to guess β.

This security game correctly models the dealer and unprivileged users in collusion in a semi-honest setting. As the adversary is allowed to select the privileged set, the message, and generates the group token by itself, which will then be used in the ciphertext generation. A BrED scheme BrED satisfies message indistinguishability from unauthorized users ($\mathsf{cpa}_{\mathcal{U}}$) if for all ppt adversary $\mathcal{A}$,

$$\mathrm{Adv}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathcal{A},\mathsf{BrED}}(\lambda) = \left| \frac{1}{2} - \Pr\left[\mathsf{Exp}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{BrED}}(1^\lambda, \mathcal{A}) = 1\right] \right| \leq \mathsf{neg}(\lambda),$$

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

Game Description	Oracle Description
$\underline{\mathsf{Exp}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathsf{BrED}}(1^\lambda, \mathcal{A})}$	$\underline{\mathcal{O}_{\mathsf{sk}}(i)}$
$\mathcal{Q}_{\mathsf{sk}} \leftarrow \emptyset$	$\mathcal{Q}_{\mathsf{sk}} \leftarrow \mathcal{Q}_{\mathsf{sk}} \cup \{i\}$
$(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{BrED.Setup}(1^\lambda, n)$	Run $\mathsf{sk}_i \leftarrow \mathsf{BrED.KeyGen}(\mathsf{msk}, i)$
$(\Gamma_S, k, M_0, M_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sk}}(\cdot)}(\mathsf{pp})$, for $S \subseteq [n]$ and $\lvert M_0\rvert = \lvert M_1\rvert$	Return sk_i
where $\Gamma_S \leftarrow \mathsf{BrED.GroupGen}(\mathsf{pp}, k, S)$	
Sample $\beta \xleftarrow{\$} \{0,1\}$, $\mathsf{CT}_{S,\beta} \leftarrow \mathsf{BrED.Enc}(\mathsf{pp}, \Gamma_S, M_\beta)$	
$\beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sk}}(\cdot)}(\mathsf{pp}, \mathsf{CT}_{S,\beta})$	
Return 1 if $(\beta = \beta')$ AND $(S \cap \mathcal{Q}_{\mathsf{sk}} = \emptyset)$.	

Figure 4: Adaptive Message Indistinguishability from Unauthorized Users of BrED

Intuitively $\mathsf{cpa}_{\mathcal{U}}$ is a strictly stronger security requirement than $\mathsf{cpa}_{\mathcal{X}}$ as the $\mathsf{cpa}_{\mathcal{U}}$ adversary can generate the group token and do secret key queries of unprivileged users. We prove this in two steps:

1. For all ppt adversary $\mathcal{A}$ against $\mathsf{cpa}_{\mathcal{X}}$, there exists a ppt adversary $\mathcal{B}$ against $\mathsf{cpa}_{\mathcal{U}}$ s.t.
$$\mathrm{Adv}^{\mathsf{cpa}_{\mathcal{X}}}_{\mathcal{A},\mathsf{BrED}}(\lambda) \leq \mathrm{Adv}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathcal{B},\mathsf{BrED}}(\lambda).$$
2. There exists a BrED scheme that achieves $\mathsf{cpa}_{\mathcal{X}}$ security but is not secure in $\mathsf{cpa}_{\mathcal{U}}$.

This ensures that the following theorem is indeed correct.

Theorem 1. $\mathsf{cpa}_{\mathcal{U}}$ *is a strictly stronger security requirement than* $\mathsf{cpa}_{\mathcal{X}}$*.*

Proof. Theorem 1 follows from the following two lemmas.

Lemma 1. *For all* ppt *adversary* $\mathcal{A}$ *against* $\mathsf{cpa}_{\mathcal{X}}$*, there exists a* ppt *adversary* $\mathcal{B}$ *against* $\mathsf{cpa}_{\mathcal{U}}$ *s.t.* $\mathrm{Adv}^{\mathsf{cpa}_{\mathcal{X}}}_{\mathcal{A},\mathsf{BrED}}(\lambda) \leq \mathrm{Adv}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathcal{B},\mathsf{BrED}}(\lambda)$.

Proof. Given an efficient adversary $\mathcal{A}$ for $\mathsf{cpa}_{\mathcal{X}}$ we construct an efficient adversary $\mathcal{B}$ for $\mathsf{cpa}_{\mathcal{U}}$.

- **Setup**: The $\mathsf{cpa}_{\mathcal{U}}$ challenger runs $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}$ gives pp to $\mathcal{B}$ who forwards it to $\mathcal{A}$.

- **Challenge**: $\mathcal{A}$ gives (S, M_0, M_1) to $\mathcal{B}$ as its challenge. $\mathcal{B}$ runs $\Gamma_S \leftarrow \mathsf{GroupGen}(\mathsf{pp}, S)$ and forwards (Γ_S, M_0, M_1) to the $\mathsf{cpa}_{\mathcal{U}}$ challenger who samples $\beta \xleftarrow{\$} \{0, 1\}$ and returns $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pp}, \Gamma_S, M_\beta)$ to $\mathcal{B}$. $\mathcal{B}$ forwards the challenge ciphertext ct to $\mathcal{A}$.

- **Guess**: When $\mathcal{A}$ outputs guess bit β', $\mathcal{B}$ forwards it to the $\mathsf{cpa}_{\mathcal{U}}$ challenger.

It is easy to see that if $\mathcal{A}$ guesses β correctly, so does $\mathcal{B}$. Thus, any BrED scheme that is secure in the $\mathsf{cpa}_{\mathcal{U}}$ model, is also secure in the $\mathsf{cpa}_{\mathcal{X}}$ model. □

Lemma 2. *There exists a BrED scheme that is secure in* $\mathsf{cpa}_{\mathcal{X}}$ *but is not secure in* $\mathsf{cpa}_{\mathcal{U}}$*.*

To show this, we construct a BrED scheme that is $\mathsf{cpa}_{\mathcal{X}}$ secure but not $\mathsf{cpa}_{\mathcal{U}}$ secure. This pathological construction is done using a modified BE scheme. This modified version of BE is basically a standard BE with an additional algorithm Verify that takes a ciphertext ct and a number k, returns 1 when the size of the privileged set w.r.t. ct is $|S| \leq k$. There exists public key BE constructions [Del07, GW09, GLR18] (where ct provides S in plain) and anonymous BE schemes [LG18] (where $|\mathsf{ct}|$ depends on $|S|$), which could support an additional algorithm Verify to check if $|S| \leq k$.

Proof. Here, we assume existence of a variant of broadcast encryption $(\mathsf{BE}) = (\mathsf{BE.Setup}, \mathsf{BE.KeyGen}, \mathsf{BE.Verify}, \mathsf{BE.Enc}, \mathsf{BE.Dec})$ where the ciphertext $\mathsf{ct} = (\mathsf{Hdr}_S, \kappa_S)$ where Hdr_S is header and κ_S is session key and given a ciphertext Verify can check the size of the set that is encrypted. We construct a BrED as follows:

- $\mathsf{Setup}(1^\lambda, n)$: $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{BE.Setup}(1^\lambda, n)$.
- $\mathsf{KeyGen}(\mathsf{msk}, i)$: $\mathsf{sk}_i \leftarrow \mathsf{BE.KeyGen}(\mathsf{msk}, i)$.
- $\mathsf{GroupGen}(\mathsf{pp}, S)$: $\Gamma_S = (\mathsf{Hdr}_S, \kappa_S) \leftarrow \mathsf{BE.Enc}(\mathsf{pp}, S, 1)$ where $\mathsf{BE.Enc}$ encrypts message 1 for the set S.
- $\mathsf{Verify}(\mathsf{pp}, \Gamma_S, k)$: $b \leftarrow \mathsf{BE.Verify}(\mathsf{pp}, \Gamma_S, k)$.
- $\mathsf{Enc}(\mathsf{pp}, \Gamma_S, M)$: $\mathsf{CT}_S = (\mathsf{Hdr}_S, M \cdot \kappa_S)$.
- $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_i, (S, \mathsf{CT}_S))$: $M \leftarrow \mathsf{BE.Dec}(\mathsf{sk}_i, (S, \mathsf{CT}_S))$.

It is easy to see that the above BrED is $\mathsf{cpa}_{\mathcal{X}}$ secure due to $\mathsf{ind\text{-}cpa}$ security of the underlying BE. We now show that the above BrED is not $\mathsf{cpa}_{\mathcal{U}}$ secure. Indeed, that is true since the $\mathsf{cpa}_{\mathcal{U}}$ adversary who computed $\Gamma_S = (\mathsf{Hdr}_S, \kappa_S)$ can easily retrieve M from the given ciphertext $\mathsf{CT}_S = (\mathsf{Hdr}_S, M \cdot \kappa_S)$. □

Thus we conclude that $\mathsf{cpa}_{\mathcal{U}}$ is a strictly stronger security requirement than $\mathsf{cpa}_{\mathcal{X}}$. □

4 Constant BrED: Constant-size Ciphertext and Key

In this section, we propose our construction of broadcast encryption with dealership protocol. This construction uses the core idea of the broadcast encryption of [GLR18] but makes clever modifications to accomodate a dealer. Looking ahead, we have proved our construction secure in the generic group model. We refer to this construction as BrED.

4.1 Construction

We present our construction next. Similar to the construction of [GLR18], BrED achieves constant-size secret key and constant-size ciphertext. However, BrED achieves adaptive secure message indistinguishability from unprivileged users in the generic group model. Informally speaking, the generic model of security helped us modify the construction of [GLR18] to use a prime order pairing group and also helped us in achieving adaptive message indistinguishability.

$\underline{\mathsf{Setup}(1^\lambda, n)}$

1: $\mathcal{PG} = (p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, g, h, e) \leftarrow \mathsf{PBGen}(1^\lambda)$
2: $\alpha, \gamma \xleftarrow{\$} \mathbb{Z}_p$ s.t. $u_i = h^{\alpha^i}$ for $i \in [n]$.
3: $\mathsf{msk} = (\alpha, \gamma, g, h)$
4: Publish $\mathsf{pp} = (g, g^\alpha, \ldots, g^{\alpha^n}, g^\gamma, u_1, \ldots, u_n, e(g,h)^\gamma)$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, x_i)}$

1: $\mathsf{sk}_{x_i} = h^{\frac{\gamma}{(\alpha + x_i)}}$

$\underline{\mathsf{GroupGen}(\mathsf{pp}, S, k)}$

1: Parse $S = \{ID_1, \ldots, ID_{k'}\}$ where $k' \leq k \leq n$
2: Define $P_S(z) = \prod_{y \in S}(z + y) = b_0 + b_1 z + \ldots + b_{k'} z^{k'}$
3: $s \xleftarrow{\$} \mathbb{Z}_N$
4: Output $(k, \Gamma_S) = (k, \omega_1, \omega_2, \omega_3, \omega_4)$ where
$\omega_1 = g^{\gamma s}$ $\quad \omega_2 = g^{s P_S(\alpha)}$
$\omega_3 = g^{\alpha^{n-k} s P_S(\alpha)}$ $\quad \omega_4 = e(g,h)^{\gamma s}$

$\underline{\mathsf{Verify}(\mathsf{pp}, \Gamma_S, k)}$

1: If $e(\omega_2, h^{\alpha^n}) = e(\omega_3, h^{\alpha^k})$, output 1
2: Otherwise output 0

$\underline{\mathsf{Enc}(\mathsf{pp}, \Gamma_S, M)}$

1: $r \xleftarrow{\$} \mathbb{Z}_N$
2: Output $\mathsf{CT}_S = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_0) = (\omega_1^r, \omega_2^r, M \cdot \omega_4^r)$

$\underline{\mathsf{Dec}(\mathsf{pp}, (\mathsf{sk}, x), (\mathsf{CT}_S, S))}$

1: Parse $\mathsf{CT}_S = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_0) = (g^{\gamma t}, g^{t P_S(\alpha)}, M \cdot e(g,h)^{\gamma t})$.
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 $A = e(\mathsf{ct}_2, \mathsf{sk}) \cdot e\left(\mathsf{ct}_1, \prod_{i \in [1, k-1]} u_i^{a_i}\right)^{-1}$.
4: Output $\frac{\mathsf{ct}_0}{A^{a_0^{-1}}}$.

Figure 5: Our BrED construction

Correctness. We parse $\mathsf{CT}_S = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_0) = (g^{\gamma t}, g^{t P_S(\alpha)}, M \cdot e(g,h)^{\gamma t})$ where $t = s.r$ for s being the randomness used in GroupGen and r is used by Enc to re-randomize the group token Γ_S. Next we evaluate the correctness of the Dec algorithm. To do so, we first evaluate the value of A in the Dec algorithm.

$$e(\mathsf{ct}_2, \mathsf{sk}) = e\left(g^{tP_S(\alpha)}, h^{\frac{\gamma}{(\alpha+x)}}\right) = e(g,h)^{\gamma t P_{S\setminus\{x\}}(\alpha)}$$

$$e\left(\mathsf{ct}_1, \prod_{i\in[1,k-1]} u_i^{a_i}\right) = e\left(\mathsf{ct}_1, \prod_{i\in[1,k-1]} h^{a_i\alpha^i}\right) = e(g^{\gamma t}, h^{P_{S\setminus\{x\}}(\alpha)-a_0})$$
$$= e(g,h)^{\gamma t P_{S\setminus\{x\}}(\alpha)} e(g,h)^{-a_0\gamma t}$$

Thus, $A = e(g,h)^{a_0\gamma t}$ and $\frac{\mathsf{ct}_0}{A^{a_0^{-1}}} = \frac{M\cdot e(g,h)^{\gamma t}}{e(g,h)^{\gamma t}} = M$.

4.2 Technical Overview

Our type-3 pairing-based BrED construction is inspired by Gong et al.'s broadcast encryption scheme [GLR18]. We use a pairing map $e : \mathbb{G}\times\mathbb{H} \to \mathbb{G}_T$, where g and h are generators of $\mathbb{G}$ and $\mathbb{H}$ respectively. For a user x, the secret key is defined as $\mathsf{sk} = h^{\frac{\gamma}{(\alpha+x)}}$, utilizing h from the public key and α, γ from the master secret key. The group token $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4)$ contains an encryption of the message 1 with respect to a privileged set S, following [GLR18]. Importantly, the encryption function is randomizable and multiplicatively homomorphic with respect to messages. Our BrED encryption randomizes the group token to $\Gamma_S^r = (\omega_1^r, \omega_2^r, \omega_4^r)$ and then homomorphically multiplies the message, resulting in a ciphertext aligned with [GLR18]. Decryption follows the procedure defined in [GLR18].

The difference lies in the fact that Gong et al. [GLR18] used the component of the master secret key γ for achieving selective security (using the Deja Q framework of [Wee16]), whereas we use it for group privacy. Specifically, the DDH instance $(g, g^\gamma, g^s, g^{\gamma s})$ allows us to simulate components of a group token without revealing the privileged set, ensuring group privacy (see Theorem 2). To ensure that the privileged set is upper bounded by the declared value k (i.e., $|S| \le k \le n$), our verification function checks if $(g^{\alpha^{n-k}}, \omega_3)$ is in the span of (g, ω_2). Using $\{h^{\alpha^i} \in \mathbb{H}\}_{i\in[n]}$, our verification function precisely checks if $e(\omega_2, h^{\alpha^n}) = e(\omega_3, h^{\alpha^k})$. We achieve maximum user accountability by ensuring that $g^{\alpha^{n+i}}$ for $i \ge 1$ is unavailable. Finally, we modify the CPA security of [GLR18] to ensure that no collaboration between unprivileged users and the dealer can retrieve any new information about the encrypted message.

4.3 Security

In this section, we prove that our construct BrED described in Figure 5 achieves all three security notions described in this work.

4.3.1 Group Privacy

Theorem 2. *Let there exists a* ppt *adversary* $\mathcal{A}$ *breaking the group privacy of* BrED *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* $\mathbb{G}$*, where* $\mathcal{PG} = (p, g, h, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) \leftarrow \mathsf{PBGen}(1^\lambda)$ *such that* $\mathbb{G} = \langle g\rangle$ *and* $\mathbb{H} = \langle h\rangle$*.*

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 problem instance (g, h, g^a, g^{ab}, Z) for $a, b \xleftarrow{\$} \mathbb{Z}_p$, where Z is either g^b or any random element form $\mathbb{G}$, $\mathcal{B}$ does the following:

- Samples $\alpha \xleftarrow{\$} \mathbb{Z}_p$.
- Implicitly sets $\gamma = a$ and $s = b$.
- Publishes the public key $\mathsf{pp} = (g, g^{\alpha}, \dots, g^{\alpha^n}, g^a, h^{\alpha}, \dots, h^{\alpha^n}, e(g,h)^a)$.
- Given pp, $\mathcal{A}$ outputs two sets (S_0, S_1) such that $|S_0| = |S_1| = k' < n$.
- $\mathcal{B}$ then chooses $\beta \xleftarrow{\$} \{0,1\}$, and returns the token $\Gamma_{S_\beta} = (\omega_1, \omega_2, \omega_3, \omega_4)$ as following:

$$\begin{array}{ll} \omega_1 = g^{ab} & \omega_2 = Z^{P_{S_\beta}(\alpha)} \\ \omega_3 = Z^{\alpha^{n-k} P_{S_\beta}(\alpha)} & \omega_4 = e(\omega_1, h) \end{array}.$$

- $\mathcal{A}$ outputs β'.
- If $\beta' = \beta$, then $\mathcal{B}$ outputs 1 else 0.

As α is chosen uniformly at random, the public key pp is properly distributed. Notice that if $Z = g^b$ then Γ_{S_β} 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 $\mathbb{G}$.

Then, the advantage of $\mathcal{B}$ in the DDH game is the same as that of adversary $\mathcal{A}$ guessing β with the probability of anything other than guessing randomly. So, if adversary $\mathcal{A}$ can win and break the group privacy of BrED, then adversary $\mathcal{B}$ can win the DDH game. Thus, breaking the group privacy is equivalent to breaking the DDH assumption in $\mathbb{G}$. □

4.3.2 Maximum User of Accountability

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 $\sigma_T : \mathbb{G}_T \to \{0,1\}^{m_T}$ w.l.o.g. $m_G \le m_H \le m_T$.

Theorem 3. *Let $\mathcal{A}$ be a* ppt *algorithm that acts as an adversary for the maximum user of accountability security of* BrED *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 that the advantage of $\mathcal{A}$ in the* maximum user of accountability *security game of* BrED *is at most $\mathcal{O}(\frac{m^2}{p})$.*

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 3 variables α, γ, s and $\sigma_{b,i}$ are strings from $\{0,1\}^{m_b}$, where $b \in \{G, H, T\}$. In this proof, we use $b \in \{G, H, T\}$ in the subscript to denote the respective groups.

Simulating pp. 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,

- for the group $\mathbb{G}$,
  - 1, α, …, α^n and γ are assigned to $f_{G,0}$, $f_{G,1}$, …, $f_{G,n}$, $f_{G,n+1}$,
- for the group $\mathbb{H}$,
  - α, …, α^n are assigned to $f_{H,1}$, $f_{H,2}$, …, $f_{H,n}$,
- for the group $\mathbb{G}_T$,

– γ is assigned to $f_{T,0}$,

respectively. A uniform random string σ_{b,ψ_b} is chosen from $\{0,1\}^{m_b}$ without repetition for each f_{b,ψ_b}, and the pair $(f_{b,\psi_b}, \sigma_{b,\psi_b})$ is appended to the list L_b, where $b \in \{G, H, T\}$. The simulator $\mathcal{C}$ publishes all the σ_{b,ψ_b} chosen as the public key so far (in appropriate order).

We assume that the adversary $\mathcal{A}$ queries the oracles on strings previously obtained from $\mathcal{C}$. 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 multiplication or division, $\mathcal{C}$ 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 $\sigma_{G,\psi_G} \xleftarrow{\$} \{0,1\}^{m_G} \setminus \{\sigma_{G,0}, \sigma_{G,1}, \ldots, \sigma_{G,\psi_G - 1}\}$, adds $(f_{G,\psi_G}, \sigma_{G,\psi_G})$ 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 $\sigma_{G,i}$ and $f_{H,j}$ in L_H corresponding to $\sigma_{H,j}$ respectively and computes $f_{T,\psi_T} = f_{G,i} \cdot f_{H,j}$. 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 $\sigma_{T,\psi_T} \xleftarrow{\$} \{0,1\}^{m_T} \setminus \{\sigma_{T,0}, \sigma_{T,1}, \ldots, \sigma_{T,\psi_T - 1}\}$, adds $(f_{T,\psi_T}, \sigma_{T,\psi_T})$ to L_T, returns σ_{T,ψ_T} to $\mathcal{A}$ and increments ψ_T by one.

Simulating group token Γ_{S^*}. 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, σ''_G), $(f'''_G, \sigma'''_G) \in L_G$, and $(f'_T, \sigma'_T) \in L_T$. Here, k is the maximal size of the privileged set S^*, that is $|S^*| = k' \leq k \leq n$. 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 $\mathcal{A}$. Precisely,

$$f'_G = \gamma s \qquad f''_G = sP_{S^*}(\alpha)$$
$$f'''_G = s\alpha^{n-k} P_{S^*}(\alpha) \qquad f'_T = \gamma s.$$

Here, $P_{S^*}(\alpha) = \prod_{y \in S^*} (\alpha + y)$ is a polynomial of degree at most n. The generic group model ensures that $\mathcal{C}$ can verify

$$\begin{aligned} f'_T &= f'_G \\ f'''_G &= f''_G \cdot \alpha^{n-k} \\ f''_G \cdot \gamma \cdot {f'_T}^{-1} &\in \mathsf{Span}(f_{G,0}, f_{G,1}, \ldots, f_{G,n}). \end{aligned}$$

Let $\mathbf{v} = (\alpha, \gamma)$ denote the vector consisting of variables over which the polynomials are defined. Now the simulator chooses at random $\alpha^*, \gamma^* \xleftarrow{\$} \mathbb{Z}_p$. Let $\mathbf{v}^* = (\alpha^*, \gamma^*)$. $\mathcal{C}$ assigns $\mathbf{v}^*$ to the variables of $\mathbf{v}$. 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 will try to bound the probability of Bad. The simulation is perfect if Bad does not happen. Let us assume $\mathcal{A}$ has generated his challenge for the set S^*, with $|S^*| > k$. We argue if Bad does not happen, $\mathcal{A}$ has no advantage in generating Γ_{S^*} from the queries it made. 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 shall be greater than n. Notice that, α^{n+i} for $i \geq 1$ 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 pp. Thus, it only has the option to do it on its own. The probability that it guesses f'''_G having no info about $g^{\alpha^{n+i}}$ is negligible.

We want to bound the Bad probability. This is where we utilize the DLSZ-lemma (see Section 2.2.2). 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 the polynomial becomes zero at random $\mathbf{v}^*$ with probability at most $(n+1)/p$. For fixed i, j, $f_{H,i} - f_{H,j}$ is a polynomial of degree at most n. Hence, the polynomial becomes 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 $(2n+1)$, hence the polynomial becomes zero at random $\mathbf{v}^*$ with probability at most $(2n+1)/p$.

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, $\psi_G + \psi_H + \psi_T \leq m + 2n + 3$. Then,

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

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 the probability that $\mathcal{A}$ wins is at most $\mathcal{O}(m^2/p)$.

Now to complete the proof, let E be an event that $\mathcal{A}$ produces Γ_{S^*} st $|S^*| > k$, and M be an event that $\mathsf{Verify}(\Gamma_{S^*}, k)$ outputs 1. So, $\Pr[E \wedge M] \leq \Pr[E] \leq (m+2n+3)^2 \cdot \frac{2n+1}{p}$. Thus,

$$\begin{aligned}\Pr[\mathsf{Verify}(\Gamma_{S^*}, k) \to 1 \wedge |S^*| > k] &\leq (m+2n+3)^2 \frac{2n+1}{p} \\ &= \mathcal{O}(m^2/p).\end{aligned}$$

□

Remark 1. We here justify the generic group model for the security proof of Maximum User of Accountability. We chose to give the proof in such an idealized model as simulation of the group-token turned out to be quite difficult if S^* is not available. This situation is in keeping with the security definition of *maximum user of accountability* in Section 2.4.2 unlike the existing works where the adversary gave away S^* contradicting their own definitions. We give the complete discussion in Appendix 3.2.

4.3.3 Message Indistinguishability from Unauthorized Users

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 $\sigma_T : \mathbb{G}_T \to \{0,1\}^{m_T}$ w.l.o.g. $m_G \leq m_H \leq m_T$.

Theorem 4. *Let $\mathcal{A}$ be a* ppt *adversary against the message indistinguishability from the unauthorized user security game of* BrED *in the generic group model. Let n be any* *natural number that is the maximal size of the set of receivers per broadcast. Let $\mathcal{A}$ make ℓ many secret key queries. Also, let m be a bound on the total number of group elements $\mathcal{A}$ receives from queries. Then the advantage of $\mathcal{A}$ in the* message indistinguishability from unauthorized users *security game of* BrED *is at most*

$$\mathsf{Adv}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathcal{A},\mathsf{BrED}} \leq (m+\ell+2n+3)^2 \cdot \frac{2n+3}{2p}.$$

Proof. We first make slight changes to the original security game of $\mathsf{cpa}_{\mathcal{U}}$. In $\mathsf{cpa}_{\mathcal{U}}$ security game, the adversary provides two messages M_0, M_1 and the challenge ciphertext has a component $M_0 \cdot e(g,h)^{\gamma sr}$ or $M_1 \cdot e(g,h)^{\gamma sr}$. For ease of proving, we modified the security game, and the adversary has to decide whether the challenge ciphertext is $M \cdot e(g,h)^{\gamma sr}$ or $e(g,h)^{\theta}$, for some $\theta \xleftarrow{\$} \mathbb{Z}_p$. Instead of sampling $\beta \xleftarrow{\$} \{0,1\}$ and providing the adversary $M_\beta \cdot e(g,h)^{\gamma sr}$ like in $\mathsf{cpa}_{\mathcal{U}}$ security game, we provide adversary y_β, where $y_0 = M \cdot e(g,h)^{\gamma sr}$ and $y_1 = e(g,h)^{\theta}$. The adversary outputs his guess β'. The adversary wins if $\beta' = \beta$. It is easy to see that these two games are identical. This kind of technique has commonly been used in literature [BSW07]. So we prove our security in the modified setting.

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 $\psi_G + \psi_H + \psi_T = \psi + \ell + 2n + 3$ holds. Here $f_{*,*}$ are multivariate polynomials over 5 variables α, γ, s, r, θ and $\sigma_{b,i}$ are strings from $\{0,1\}^{m_b}$, where $b \in \{G,H,T\}$. In this proof, we use $b \in \{G,H,T\}$ in the subscript to denote the respective groups.

Simulating pp. 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,

- for the group $\mathbb{G}$,
  - $1, \alpha, \ldots, \alpha^n$ and γ are assigned to $f_{G,0}, f_{G,1}, \ldots, f_{G,n}, f_{G,n+1}$,
- for the group $\mathbb{H}$,
  - $\alpha, \ldots, \alpha^n$ are assigned to $f_{H,1}, f_{H,2}, \ldots, f_{H,n}$,
- for the group $\mathbb{G}_T$,
  - γ is assigned to $f_{T,0}$,

respectively. A uniform random string σ_{b,ψ_b} is chosen from $\{0,1\}^{m_b}$ without repetition for each f_{b,ψ_b}, and the pair $(f_{b,\psi_b}, \sigma_{b,\psi_b})$ is appended to the list L_b, where $b \in \{G,H,T\}$. The simulator $\mathcal{C}$ publishes all the σ_{b,ψ_b} chosen as the public key so far (in appropriate order).

We assume that the adversary $\mathcal{A}$ queries the oracles on strings previously obtained from $\mathcal{C}$. 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 multiplication or division, $\mathcal{C}$ 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 $\sigma_{G,\psi_G} \xleftarrow{\$} \{0,1\}^{m_G} \setminus \{\sigma_{G,0}, \sigma_{G,1}, \ldots, \sigma_{G,\psi_G-1}\}$, adds $(f_{G,\psi_G}, \sigma_{G,\psi_G})$ 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 $\sigma_{G,i}$ and $f_{H,j}$ in L_H corresponding to $\sigma_{H,j}$ respectively and computes $f_{T,\psi_T} = f_{G,i} \cdot f_{H,j}$. 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 $\sigma_{T,\psi_T} \xleftarrow{\$} \{0,1\}^{m_T} \setminus \{\sigma_{T,0}, \sigma_{T,1}, \ldots, \sigma_{T,\psi_T-1}\}$, adds $(f_{T,\psi_T}, \sigma_{T,\psi_T})$ to L_T, returns σ_{T,ψ_T} to $\mathcal{A}$ and increments ψ_T by one.

Simulating sk. On receiving secret key query for x_i the simulator $\mathcal{C}$ assigns the polynomials, $\gamma(\alpha + x_i)^{-1}$ to f_{H,ψ_H}, samples a string $\sigma_{H,\psi_H} \xleftarrow{\$} \{0,1\}^{m_H} \setminus \{\sigma_{H,0}, \sigma_{H,1}, \ldots, \sigma_{H,\psi_H-1}\}$, adds $(f_{H,\psi_H}, \sigma_{H,\psi_H})$ to L_H, returns σ_{H,ψ_H} to $\mathcal{A}$ and increments ψ_H by one. The simulator also adds $(x_i, (f_{H,\psi_H}, \sigma_{H,\psi_H}))$ to $\mathcal{Q}_{\mathsf{sk}}$. For each secret key query, the adversary $\mathcal{A}$ gets one string from the group $\mathbb{H}$. So if $\mathcal{A}$ makes at most ℓ-many secret keys queries, then it gets ℓ many elements from $\mathbb{H}$. So $\psi_H = \ell + n$.

Simulating challenge ciphertext CT_{S^*} 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 $(f'_T, \sigma'_T) \in L_T$. Here, k is the maximal size of the privileged set S^*, that is $|S^*| = k' \leq k \leq n$. 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 $\mathcal{A}$. Precisely,

$$f'_G = \gamma s \qquad f''_G = s P_{S^*}(\alpha)$$
$$f'''_G = s\alpha^{n-k} P_{S^*}(\alpha) \qquad f'_T = \gamma s.$$

Here $P_{S^*}(\alpha) = \prod_{y \in S^*} (\alpha + y)$ is a polynomial of degree at most n. Due to natural restrictions all $y \in S$; $(y, \cdot) \notin \mathcal{Q}_{\mathsf{sk}}$, and generic group model allows the simulator $\mathcal{C}$ to check that $P_{S^*}(\alpha)$ does not involve any user y for which secret key has been queried. The generic group model ensures that $\mathcal{C}$ can verify

$$\begin{aligned} f'_T &= f'_G \\ f'''_G &= f''_G \cdot \alpha^{n-k} \\ f''_G \cdot \gamma \cdot {f'_T}^{-1} &\in \mathsf{Span}(f_{G,0}, f_{G,1}, \ldots, f_{G,n}). \end{aligned}$$

Finally, $\mathcal{C}$ samples $\beta \xleftarrow{\$} \{0,1\}$ and computes $\widetilde{f}'_G = f'_G \cdot r$, $\widetilde{f}''_G = f''_G \cdot r$ and $\widetilde{f}'_T = y_\beta$, where $y_0 = f'_T \cdot r$ and $y_1 = \theta$. $\mathcal{C}$ then adds $(\widetilde{f}'_G, \widetilde{\sigma}'_G), (\widetilde{f}''_G, \widetilde{\sigma}''_G) \in L_G$, and $(\widetilde{f}'_T, \widetilde{\sigma}'_T) \in L_T$ following the above rules on group actions. At this point, $\mathcal{C}$ returns $(\widetilde{\sigma}'_G, \widetilde{\sigma}''_G, \widetilde{\sigma}'_T)$ to $\mathcal{A}$ who returns β'.

Let $\mathbf{v} = (\alpha, \gamma, r, \theta)$ denote the vector consisting of variables over which the polynomials are defined. Now the simulator chooses at random $\alpha^*, \gamma^*, r^*, \theta^* \xleftarrow{\$} \mathbb{Z}_p$. Let $\mathbf{v}^* = (\alpha^*, \gamma^*, r^*, \theta^*)$. $\mathcal{C}$ assigns $\mathbf{v}^*$ to the variables of $\mathbf{v}$. 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 of all, 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. Recall $\mathcal{A}$ has access to all the lists (L_G, L_H, L_T) and gets $(\widetilde{\sigma}'_G, \widetilde{\sigma}''_G, \widetilde{\sigma}'_T)$ as its challenge where $(\widetilde{f}'_G, \widetilde{\sigma}'_G), (\widetilde{f}''_G, \widetilde{\sigma}''_G) \in L_G$, and $(\widetilde{f}'_T, \widetilde{\sigma}'_T) \in L_T$. Observe that, $\widetilde{f}'_G, \widetilde{f}''_G, \widetilde{f}'_T$ are respectively $\gamma^* sr^*$, $\gamma^* sr^* \cdot P_{S^*}(\alpha^*)$ and $\gamma^* sr^*$, where $\widetilde{f}'_T = \gamma^* sr^*$ in group $\mathbb{G}_T$ is our target polynomial.

It is clear that $\widetilde{f}'_T$ is a three-degree polynomial defined in the group $\mathbb{G}_T$. To compute such $\widetilde{f}'_T$, we mention that $\mathcal{A}$ can use the polynomial lists L_G, challenge polynomials $\widetilde{f}'_G$, $\widetilde{f}''_G$, and the polynomials from group token f'_G, f''_G to pair with the polynomials of L_H. The list of polynomials available to $\mathcal{A}$ is the following.

- Polynomials from group $\mathbb{G}$,
  - 1, α^*, …, $(\alpha^*)^n$ and γ^* (from public parameters),
  - $\gamma^* s$, $\gamma^* s P_{S^*}(\alpha^*)$ (from the group token),
  - $\gamma^* sr^*$, $\gamma^* sr^* P_{S^*}(\alpha^*)$ (from the challenge cipher-text).
- Polynomials from group $\mathbb{H}$,
  - (α^*), …, $(\alpha^*)^n$ (from public parameters),
  - $\gamma^*(\alpha^* + x_i)^{-1}, \forall (x_i, \cdot) \in \mathcal{Q}_{\mathsf{sk}}$ (from secret key queries).
    Note that $(\alpha^* + x_i)$ does not divide $P_{S^*}(\alpha^*)$ due to natural restrictions.
- Polynomials from group $\mathbb{G}_T$,
  - γ^* (from public parameters).

The only three-degree polynomials that can be constructed combining one polynomial from L_G and one from L_H are $(\alpha^*)^3$, $(\alpha^*)^2\gamma^*$, $\alpha^*\gamma^* s$ and any linear combinations of these three polynomials. All the above-mentioned polynomials that can be constructed for the group $\mathbb{G}_T$ do not involve the challenge polynomial $\gamma^* sr^*$. Thus, the best $\mathcal{A}$ can do is to output its guess β' at random and subsequently $\Pr[\beta = \beta' : \neg\mathsf{Bad}] = 1/2$.

We would like to bound the probability of Bad. Here we use DLSZ-lemma (see Section 2.2.2). 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 $x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p$, 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 $n+3$. Hence, the polynomial becomes zero at random $\mathbf{v}^*$ with probability at most $(n+3)/p$. For fixed i, j, $f_{H,i} - f_{H,j}$ is a polynomial of degree at most n. Hence, the polynomial becomes 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 $(2n+3)$, hence the polynomial becomes zero at random $\mathbf{v}^*$ with probability at most $(2n+3)/p$. 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, $\psi_G + \psi_H + \psi_T \leq m + \ell + 2n + 3$. Then,

$$
\begin{aligned}
\Pr[\mathsf{Bad}] &\leq \binom{\psi_G}{2} \frac{n+3}{p} + \binom{\psi_H}{2} \frac{n}{p} + \binom{\psi_T}{2} \frac{2n+3}{p} \\
&\leq (m + \ell + 2n + 3)^2 \cdot \frac{4n+6}{2p} \\
&\leq (m + \ell + 2n + 3)^2 \cdot \frac{2n+3}{p}.
\end{aligned}
$$

Now, a simple argument shows that,

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

Also,

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

These two results were combined to give us

$$\begin{aligned}\mathsf{Adv}^{\mathsf{cpa}_{\mathcal{U}}}_{\mathcal{A},\mathsf{BrED}} = \left|\Pr[\beta=\beta'] - \frac{1}{2}\right| &\leq \frac{\Pr[\mathsf{Bad}]}{2} \\ &\leq (m+\ell+2n+3)^2\cdot\frac{2n+3}{2p}.\end{aligned}$$

□

5 Conclusion

In this paper, we have shown the limitations of the existing works. Precisely, we found security issues in all of them, rendering them insecure. We have shown that all previously considered security definitions of BrED do not model the real world, and we corrected them. We propose new constructions that are secure in the newly proposed security model. Furthermore, we also achieve constant size ciphertext and secret key, which make our scheme ready to be deployed in real life. In this work, we only have considered *semi-honest* dealer and allowed collusion with unprivileged users. Due to the highly interactive nature of BrED, we could only achieve some security in the generic group model. We suggest getting standard assumption-based proof in a fully malicious dealer model as possible for future work.

Acknowledgement

We would like to thank Dr. Debrup Ckahraborty for the discussions and insightful comments on this work. We thank anonymous reviewers of Communications in Cryptology for their comments.

References

[AD16] Kamalesh Acharya and Ratna Dutta. Secure and efficient construction of broadcast encryption with dealership. In Liqun Chen and Jinguang Han, editors, *Provable Security - 10th International Conference, ProvSec 2016, Nanjing, China, November 10-11, 2016, Proceedings*, volume 10005 of *Lecture Notes in Computer Science*, pages 277–295, 2016. doi:10.1007/978-3-319-47422-9_16.

[AD17] Kamalesh Acharya and Ratna Dutta. Recipient revocable broadcast encryption schemes without random oracles. In Howon Kim and Dong-Chan Kim, editors, *Information Security and Cryptology - ICISC 2017 - 20th International Conference, Seoul, South Korea, November 29 - December 1, 2017, Revised Selected Papers*, volume 10779 of *Lecture Notes in Computer Science*, pages 191–213. Springer, 2017. doi:10.1007/978-3-319-78556-1_11.

[AD21] Kamalesh Acharya and Ratna Dutta. Constructing provable secure broadcast encryption scheme with dealership. *J. Inf. Secur. Appl.*, 58:102736, 2021. URL: https://doi.org/10.1016/j.jisa.2020.102736, doi:10.1016/J.JISA.2020.102736.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2005. doi:10.1007/11535218_16.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*, pages 321–334. IEEE Computer Society, 2007. doi:10.1109/SP.2007.11.

[Del07] Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*, pages 200–215. Springer, 2007. doi:10.1007/978-3-540-76900-2_12.

[DL78] Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978. doi:10.1016/0020-0190(78)90067-4.

[Duc10] Léo Ducas. Anonymity from asymmetry: New constructions for anonymous HIBE. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*, volume 5985 of *Lecture Notes in Computer Science*, pages 148–164. Springer, 2010. doi:10.1007/978-3-642-11925-5_11.

[FN93] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993. doi:10.1007/3-540-48329-2_40.

[GLR18] Junqing Gong, Benoît Libert, and Somindu C. Ramanna. Compact IBBE and fuzzy IBE from simple assumptions. In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 563–582. Springer, 2018. doi:10.1007/978-3-319-98113-0_30.

[GSP+16] Clémentine Gritti, Willy Susilo, Thomas Plantard, Kaitai Liang, and Duncan S. Wong. Broadcast encryption with dealership. *Int. J. Inf. Sec.*, 15(3):271–283, 2016. URL: https://doi.org/10.1007/s10207-015-0285-x, doi:10.1007/S10207-015-0285-X.

[GW09] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009. doi:10.1007/978-3-642-01001-9_10.

[KLEL17] Joon Sik Kim, Young Kyung Lee, Jieun Eom, and Dong Hoon Lee. Recipient revocable broadcast encryption with dealership. In Howon Kim and Dong-Chan Kim, editors, *Information Security and Cryptology - ICISC 2017 - 20th International Conference, Seoul, South Korea, November 29 - December 1, 2017, Revised Selected Papers*, volume 10779 of *Lecture Notes in Computer Science*, pages 214–228. Springer, 2017. doi:10.1007/978-3-319-78556-1_12.

[LG18] Jiangtao Li and Junqing Gong. Improved anonymous broadcast encryptions - tight security and shorter ciphertext. In Bart Preneel and Frederik Vercauteren, editors, *Applied Cryptography and Network Security - 16th International Conference, ACNS 2018, Leuven, Belgium, July 2-4, 2018, Proceedings*, volume 10892 of *Lecture Notes in Computer Science*, pages 497–515. Springer, 2018. doi:10.1007/978-3-319-93387-0_26.

[RCS12] Somindu C. Ramanna, Sanjit Chatterjee, and Palash Sarkar. Variants of waters' dual system primitives using asymmetric pairings - (extended abstract). In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, volume 7293 of *Lecture Notes in Computer Science*, pages 298–315. Springer, 2012. doi:10.1007/978-3-642-30057-8_18.

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1997. doi:10.1007/3-540-69053-0_18.

[Wee16] Hoeteck Wee. Déjà Q: encore! un petit IBE. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 237–258. Springer, 2016. doi:10.1007/978-3-662-49099-0_9.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5_73.

A An Overview of Existing Works

For the convenience of readers, in this section, we describe all the existing [AD16, AD17, KLEL17, AD21] constructions briefly. None of the Works did follow a consistent notation. In the following, we bring them into a common notation that is consistent throughout this work. We note that all the existing papers [AD16, AD17, KLEL17, AD21] are instantiated in the bilinear pairing group system. Therefore, we assume the existence of $\mathcal{BG} = (p, g, \mathbb{G}, \mathbb{G}_T, e)$ 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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is the bilinear mapping. Let n denote the maximal number of receivers per broadcast. 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 the maximum number of revocations 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 the respective papers [AD16, AD17, KLEL17, AD21].

A.1 Brief Description of BrED Scheme of [AD16]

Authors of [AD16] presented their construction from a key encapsulation mechanism with the dealership (KEMD). The description of their scheme is as follows.

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{KEMD.Setup}(1^\lambda, n)$: 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
  $$\mathsf{pp} = \Big(\mathcal{BG}, g, g^\alpha, \ldots, g^{\alpha^n}, e(g,h), h^\alpha, H, \mathsf{ID}\Big).$$
- $(\mathsf{sk}_i) \leftarrow \mathsf{KEMD.KeyGen}(\mathsf{pp}, \mathsf{msk}, i)$: 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.
- $(\Gamma_S, k) \leftarrow \mathsf{KEMD.GroupGen}(\mathsf{pp}, S)$: For a group of users $S = \{ID_1, ID_2, \ldots, ID_{k'}\} \subseteq [n]$, generate a group token $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4)$ as,
  1. Define $P_S(x) = \prod_{ID_j \in S}(x + H(ID_j)) = \sum_{i=0}^{k'} P_i x^i$. P_i's are functions of $H(ID_j)$ for $ID_j \in S$.
  2. Sample $s \xleftarrow{\$} \mathbb{Z}_p$ and generate $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4)$ as,
  $$\omega_1 = h^{-s\alpha} \qquad \omega_3 = g^{sP_S(\alpha)}$$
  $$\omega_2 = g_{n-k}^{sP_S(\alpha)} \qquad \omega_4 = e(g,h)^s.$$
  3. Set a group threshold k for group size S where $k \geq k' = |S|$.
  4. Send S to users and publish (Γ_S, k).
- $(0/1) \leftarrow \mathsf{KEMD.Verify}(\mathsf{pp}, \Gamma_S, k)$: The verification work as following,
  $$\mathsf{KEMD.Verify}(\mathsf{pp}, \Gamma_S, k) = \begin{cases} 1, & \text{if } e(\omega_2, g^{\alpha^k}) = e(\omega_3, g^{\alpha^n}) \\ 0, & \text{otherwise.} \end{cases}$$

- $(\mathsf{Hdr}, K) \leftarrow \mathsf{KEMD.Encrypt}(\mathsf{pp}, \Gamma_S)$: Extract $(\omega_1, \omega_3, \omega_4)$ from Γ_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.

A.2 Brief Description of BrED Scheme of [AD17]

The BrED construction of [AD17] by the same set of authors is as follows.

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{BrED.Setup}(n, 1^\lambda)$:

  1. Sample $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and set,

  $$\mathsf{pp} = (\mathcal{BG}, l_0, l_0^{\alpha}, \ldots, l_0^{\alpha^n}, g, g^{\alpha}, \ldots, g^{\alpha^{n+1}}, e(g,g), e(g,l_0), \mathsf{ID});$$
  $$\mathsf{msk} = \alpha,$$

  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 pp.

- $(\mathsf{sk}_i) \leftarrow \mathsf{BrED.KeyGen}(\mathsf{pp}, \mathsf{msk}, i)$: 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,

  $$d_{1,i} = (h_i \cdot g^{r_i})^{\frac{1}{\alpha(\alpha + ID_i)}} \qquad d_{2,i} = r_i$$
  $$d_{3,i} = \left(h_i \cdot l_0^{d_{2,i}}\right)^{\frac{1}{\alpha}} \qquad \mathsf{label}_i = \left(h_i, h_i^{\alpha} \ldots, h_i^{\alpha^n}\right).$$

  Send sk_i to user i through a secure channel.

- $(\Gamma_S, k) \leftarrow \mathsf{BrED.GroupGen}(\mathsf{pp}, S)$: 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 a group token (Γ_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 generate a group token $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ as following,

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

  Send S to subscribed users via a secure channel and publish (Γ_S, k).

- $(0/1) \leftarrow \mathsf{BrED.Verify}(\mathsf{pp}, \Gamma_S, k)$: Parse $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ and checks,

  $$\mathsf{BrED.Verify}(\mathsf{pp}, \Gamma_S, k) = \begin{cases} 1, & \text{if } e(\omega_1, g^{\alpha^n}) = e(\omega_2, g^{\alpha^k}) \\ 0, & \text{otherwise.} \end{cases}$$

- $(\mathsf{ct}) \leftarrow \mathsf{BrED.Encrypt}(\mathsf{pp}, \Gamma_S, M)$: Parse $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$, sample $r \xleftarrow{\$} \mathbb{Z}_p$ and compute ciphertext as

  $$\mathsf{ct} = (ct_1, ct_2, ct_3, ct_4) = ((\omega_1^r, \omega_3^r, \omega_4^r, M \cdot \omega_5^r)).$$

A.3 Brief Description of BrED Scheme of [KLEL17]

The construction of [KLEL17] the acronym RR was used for recipient revocation. They referred to their scheme as RR-BED.

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{RR\text{-}BED.Setup}(1^\lambda, n)$: Choose $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p$, $h \xleftarrow{\$} \mathbb{G}$ and compute,

$$\begin{aligned}\mathsf{pp} =& (\mathcal{BG}, h, h^\alpha, \dots, h^{\alpha^n}, g, g^\alpha, \dots, g^{\alpha^n},\\ & g^{\alpha\beta}, \dots, g^{\alpha^{n+1}\beta}, e(g,g), e(g,h), \mathsf{ID})\\ \mathsf{msk} =& (\alpha, \beta).\end{aligned}$$

  Keep msk secret and publish pp.

- $(\mathsf{sk}_i) \leftarrow \mathsf{RR\text{-}BED.KeyGen}(\mathsf{pp}, \mathsf{msk}, i)$: 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,

$$d_{1,i} = (l_i \cdot g^{r_i})^{\frac{1}{\alpha\beta(\alpha + ID_i)}} \qquad d_{2,i} = r_i$$

$$d_{3,i} = \left(l_i h_i^{d_{2,i}}\right)^{\frac{1}{\alpha\beta}} \qquad \mathsf{label}_i = \left(l_i, l_i^\alpha \dots, l_i^{\alpha^n}\right).$$

  Send sk_i to user i through a secure channel.

- $(\Gamma_S) \leftarrow \mathsf{RR\text{-}BED.GroupGen}(\mathsf{pp}, S, k, v)$: Select a threshold value k and a group $S = \{ID_{i_1}, \dots, ID_{i_{k'}}\} \subseteq [n]$ of k' many users where $k' \leq k$ and generate a group token (Γ_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 $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ as,

$$\omega_1 = g^{s\alpha\beta P_S(\alpha)} \qquad \omega_2 = g^{s\alpha^{n-v+1}\beta P_S(\alpha)} \qquad \omega_3 = [g^{-s\alpha^i}]_{1 \leq i \leq k+1}.$$
$$\omega_4 = e(g,g)^{-s} \qquad \omega_5 = e(g,h)^s$$

  Send S to subscribed users via a secure channel and publish (Γ_S, k).

- $(0/1) \leftarrow \mathsf{RR\text{-}BED.Verify}(\mathsf{pp}, \Gamma_S, k)$: Parse $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ and check,

$$\mathsf{BrED.Verify}(\mathsf{pp}, \Gamma_S, k) = \begin{cases} 1, & \text{if } e(\omega_1, g^{\alpha^n}) = e(\omega_2, g^{\alpha^k}) \\ 0, & \text{otherwise.} \end{cases}$$

- $(\mathsf{ct}) \leftarrow \mathsf{RR\text{-}BED.Encrypt}(\mathsf{pp}, \Gamma_S, M)$: Parse $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$, sample $r \xleftarrow{\$} \mathbb{Z}_p$ and compute ciphertext as

$$\begin{aligned}\mathsf{ct} &= (ct_1, ct_2, \widehat{ct_1}, \dots, \widehat{ct_{k+1}}, ct_4)\\ &= ((\omega_1^r, \omega_3^r, \widehat{\omega_1}^r, \dots, , \widehat{\omega_{k+1}}^r, M \cdot \omega_5^r)).\end{aligned}$$

- $(\mathsf{ct}') \leftarrow \mathsf{RR\text{-}BED.Revoke}(\mathsf{ct}, R, \mathsf{pp})$: Parse ct as $(ct_1, ct_2, \widehat{ct_1}, \dots, \widehat{ct_{k+1}}, ct_4)$. Let $R = \{ID_{i_1}, \dots, ID_{i_l}\} \subseteq S$ 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 $y = t \sum_{i=0}^{l} f_i \alpha^i$ where $t = rs$ (the random coins chosen by dealer and broadcaster) and compute,

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

A.4 Brief Description of BrED Scheme of [AD21]

In the [AD21], authors proposed two BrED constructions using a symmetric bilinear pairing group where the first construction (called KEMD-I) achieves semi-static security in the standard model and the second construction (called KEMD-II) achieves adaptive security. The second construction is exactly the same as the construction in [AD17] with different variable names. We find that redundant to discuss these KEMD-II construction as the same attack that we'll show for [AD17] in the following section would hold for KEMD-II also. Here we give a brief description of their construction KEMD-I.

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{KEMD.Setup}\left(1^\lambda, n\right)$:
  1. Sample $g_1, h_1 \xleftarrow{\$} \mathbb{G}$ and $\alpha, \beta, \gamma_1 \xleftarrow{\$} \mathbb{Z}_p$
  2. Set $\widehat{g}_1 = g_1^\beta$ and $\widehat{h}_1 = h_1^\beta$
  3. Set $\mathsf{msk} \leftarrow (\alpha, h_1, \gamma_1)$ and publish

$$\mathsf{pp} \leftarrow \left(\mathcal{BG}, \{g_1^{\alpha^j}, \widehat{g}_1^{\alpha^j}\}_{j \in [0,n]}, \{\widehat{h}_1^{\alpha^k}\}_{j \in [0,n]}, g_1^{\gamma_1}, g_1^{\alpha \gamma_1} \right).$$

- $\mathsf{sk}_i \leftarrow \mathsf{KEMD.GeyGen}(\mathsf{pp}, \mathsf{msk}, i)$:
  1. Return $\mathsf{sk}_i = h_1^{\frac{\gamma_1}{\alpha + i}}$
- $(\Gamma_S, k) \leftarrow \mathsf{KEMD.GroupGen}(\mathsf{pp}, S)$: Select $S = \{ID_{i_1}, ID_{i_2}, \ldots, ID_{k'}\} \subseteq [n]$ such that $k' \leq k \leq n$ and compute
  1. Generate $A(x), A1(x), A2(x)$ as,

$$A(x) = \prod_{i \in S} (x - i) = \sum_{i=0}^{k'} A_i x^i,$$
$$A1(x) = A(x) - \prod_{i \in S} i = \sum_{i=1}^{k'} A_i x^i,$$
$$A2(x) = \frac{A1(x)}{x} = \sum_{i=1}^{k'} A_i x^{i-1},$$

where coefficients $A_i \in \mathbb{Z}_p$.

2. Generate $B(x), B1(x), B2(x)$ as,

$$B(x) = \prod_{i\in[n]\setminus S}(x-n+i) = \sum_{i=0}^{n-k'} B_i x^i,$$
$$B1(x) = B(x) - \prod_{i\in[n]\setminus S} i = \sum_{i=0}^{N-k'} B1_i x^i,$$
$$B2(x) = \frac{B1(x)}{x-n} = \sum_{i=0}^{N-k'-1} B2_i x^i,$$

where coefficients $B_i, B1_i, B2_i \in \mathbb{Z}_p$.

3. Generate $\widehat{P}(x)$ as,

$$\widehat{P}(x) = \prod_{i\in S}(x-i) \cdot \prod_{i\in[n]\setminus S}(x-n+i) = \sum_{i=0}^{n} \widehat{P}_i x^i,$$

where coefficient $\widehat{P}_i \in \mathbb{Z}_p$.

4. Let $Y = \prod_{i\in S} i$, $Z = \prod_{i\in[n]\setminus S} i$. Select $t \xleftarrow{\$} \mathbb{Z}_p^*$ and generate a group token $\Gamma_S = \left(\{\omega_i\}_{i\in[1,12]}\right)$ as following,

$$\begin{aligned} \omega_1 &= \widehat{g}_1^{A1(\alpha)t} & \omega_2 &= \widehat{g}_1^{A2(\alpha)t} \\ \omega_3 &= \widehat{g}_1^{\alpha^{n-k}A1(\alpha)t} & \omega_4 &= \widehat{g}_1^{Yt} \\ \omega_5 &= \widehat{g}_1^{\alpha^n Yt} & \omega_6 &= \widehat{g}_1^{B1(\alpha)t} \\ \omega_7 &= \widehat{g}_1^{B2(\alpha)t} & \omega_8 &= \widehat{g}_1^{Zt} \\ \omega_9 &= \widehat{g}_1^{\widehat{P}(\alpha)t} & \omega_{10} &= \widehat{g}_1^{t} \\ \omega_{11} &= g_1^{\gamma_1 t} & \omega_{12} &= g_1^{\alpha\gamma_1 t} \end{aligned}$$

5. Send S to privileged users via secure channel and public (Γ_S, k).

- $(0/1) \leftarrow \mathsf{KEMD.Verify}(\mathsf{pp}, \Gamma_S, k)$: Parse $\Gamma_S = (\omega_1, \ldots, \omega_{12}), k$ and return 1 only **if** all the following equations holds **else** return 0.

$$e\left(\omega_1, g_1^{\alpha^n}\right) = e\left(\omega_3, g_1^{\alpha^k}\right) \tag{2}$$
$$e(\omega_5, g_1) = e\left(\omega_4, g_1^{\alpha^n}\right) \tag{3}$$
$$e(\omega_2, \widehat{g}_1^{\alpha}) = e(\omega_1, \widehat{g}_1) \tag{4}$$
$$e\left(\omega_7, \widehat{g}_1^{\alpha-n}\right) = e(\omega_6, \widehat{g}_1) \tag{5}$$
$$e(\omega_4, \omega_8) = e\left(\widehat{g}_1^{\{t\prod_{i\in[n]} i\}}, \omega_{10}\right) \tag{6}$$
$$e(\omega_9, \omega_{10}) = e(\omega_1 \cdot \omega_4, \omega_6 \cdot \omega_8) \tag{7}$$

- $(\mathsf{Hdr}, K) \leftarrow \mathsf{KEMD.Encrypt}(\mathsf{pp}, \Gamma_S)$: Sample $r \xleftarrow{\$} \mathbb{Z}_p$ set the session key

$$K = e\left(\omega_{12}^r, \widehat{h}_1^{\alpha^{n-2}}\right) = e\left(g_1, \widehat{h}_1\right)^{\alpha^{n-1} s\gamma_1}$$
$$\mathsf{Hdr} = (C_1, C_2) = (\omega_9^r, \omega_{11}^r),$$

where $s = tr$ (say).

B Security Flaws of Existing Works

All previous works [AD16, AD17, KLEL17, AD21] followed a similar path to argue the security of *group privacy* and *maximum user of accountability*. The fundamental idea of the attack on group privacy is the same. However, the description of the attack changes with every scheme. Therefore, we will address the attack on group privacy individually in the following section. But to keep our presentation concise, we discuss the flaw in the security argument of the maximum user of accountability of [AD16] and omit the same discussion for [AD17, KLEL17, AD21]. In light of our discussion, readers can also appreciate the flow in all other constructions.

B.1 Description of attack of Group Privacy on [AD16]

The attack is described in the main body of the paper (See Section 3.1.1).

B.2 Description of attack of Group Privacy on [AD17]

The construction of [AD17] suffer from similar vulnerability as found in [AD16]. Precisely, the pp in [AD17] contains $(g, g^{\alpha}, \ldots, g^{\alpha^n})$. Here also adversary can easily compute $g^{P_S(\alpha)}$ using pp as $P_S(x) = \prod_{ID_{i_j} \in S} (x + ID_{i_j})$. Description of group token in this case is $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ as following,

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

Here $s \xleftarrow{\$} \mathbb{Z}_p$.

Proceeding similar to the attack of [AD16], the adversary chooses two privileged sets of equal size, S_0, S_1 and submits to the 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,

$$b = \begin{cases} 0 & \text{if } e\left(g^{P_{S_0}(\alpha)}, \omega_3\right) \times e(\omega_1, g) = 1 \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

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.

B.3 Description of Attack of Group Privacy on [KLEL17]

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 [KLEL17] contains

$$\mathsf{pp} = (\mathcal{BG}, h, h^{\alpha}, \ldots, h^{\alpha^n}, g, g^{\alpha}, \ldots, g^{\alpha^n}, g^{\alpha\beta}, \ldots, g^{\alpha^{n+1}\beta}, e(g,g), e(g,h), \mathsf{ID}).$$

Like previous two constructions here also adversary can compute $g^{\alpha\beta P_S(\alpha)}$ as $P_S(x) = \prod_{ID_{i_j} \in S} (x + ID_{i_j})$. Group tokens in their construction $\Gamma_S = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ as following,

$$\omega_1 = g^{s\alpha\beta P_S(\alpha)} \qquad \omega_2 = g^{s\alpha^{n-v+1}\beta P_S(\alpha)} \qquad \omega_3 = [g^{-s\alpha^i}]_{1 \le i \le k+1}.$$
$$\omega_4 = e(g,g)^{-s} \qquad \omega_5 = e(g,h)^s$$

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,

$$b = \begin{cases} 0 & \text{if } e\left(g^{\alpha\beta P_{S_0}(\alpha)}, \widehat{\omega_1}\right) \times e(\omega_1, g^{\alpha}) = 1 \\ 1 & \text{otherwise.} \end{cases} \qquad (9)$$

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

B.4 Description of Attack of Group Privacy on [AD21]

The authors of [AD21] has proposed two construction KEMD-I and KEMD-II. Their first construction KEMD-I is semi-static secure, whereas the second construction KEMD-II is adaptive secure. Now the second construction, *i.e.*, KEMD-II is exactly the same as the construction of [AD17]. So the attack we described for [AD17] also holds for this construction. In this section, we describe the attack on group privacy for their first construction, which is KEMD-I.

The public key of [AD21] contains,

$$\mathsf{pp} \leftarrow \left(\{g_1^{\alpha^j}, \widehat{g}_1^{\alpha^j}\}_{j \in [0,n]}, \{\widehat{h}_1^{\alpha^k}\}_{j \in [0,n]}, g_1^{\gamma_1}, g_1^{\alpha\gamma_1}\right),$$

and

$$\widehat{P}(x) = \prod_{i \in S}(x - i) \cdot \prod_{i \in [n] \setminus S}(x - n + i) = \sum_{i=0}^{n} \widehat{P}_i x^i,$$

so $\widehat{g_1}^{\widehat{P}_S(\alpha)}$ for any $S \subseteq [n]$ can be computed using public parameters. Now, for the attack, the adversary chooses two sets $S_0, S_1 \subset [n]$ of equal size and submits to the challenger. The challenger chooses $b \xleftarrow{\$} \{0,1\}$ and provide group token for Γ_{S_b}. Recall that the group token consists $\Gamma_S = \left(\{\omega_i\}_{i\in[1,12]}\right)$ where,

$$\omega_1 = \widehat{g}_1^{A1(\alpha)t} \qquad \omega_2 = \widehat{g}_1^{A2(\alpha)t}$$
$$\omega_3 = \widehat{g}_1^{\alpha^{n-k}A1(\alpha)t} \qquad \omega_4 = \widehat{g}_1^{Yt}$$
$$\omega_5 = \widehat{g}_1^{\alpha^n Yt} \qquad \omega_6 = \widehat{g}_1^{B1(\alpha)t}$$
$$\omega_7 = \widehat{g}_1^{B2(\alpha)t} \qquad \omega_8 = \widehat{g}_1^{Zt}$$
$$\omega_9 = \widehat{g}_1^{\widehat{P}(\alpha)t} \qquad \omega_{10} = \widehat{g}_1^{t}$$
$$\omega_{11} = g_1^{\gamma_1 t} \qquad \omega_{12} = g_1^{\alpha\gamma_1 t}.$$

So the adversary computes $e\left(\widehat{g_1}^{P_{S_0}(\alpha)}, \omega_{10}\right)$ which is $e\left(\widehat{g_1}^{P_{S_0}(\alpha)}, \widehat{g_1}^{t}\right) = e\left(\widehat{g_1}, \widehat{g_1}\right)^{tP_{S_0}(\alpha)}$. And on the other hand adversary also computes $e\left(\omega_9, \widehat{g_1}\right)$ which is $e\left(\widehat{g_1}^{tP_{S_b}(\alpha)}, \widehat{g_1}\right) = e\left(\widehat{g_1}, \widehat{g_1}\right)^{tP_{S_b}(\alpha)}$. So adversary checks if $e\left(\widehat{g_1}^{P_{S_0}(\alpha)}, \omega_{10}\right) = e\left(\omega_9, \widehat{g_1}\right)$ then $b = 0$ else 1.

B.5 Issue with Group Privacy Security Argument

Recall from Section 2.4.1, the *group privacy* security game of BrED allows an adversary to submit two privileged sets S_0 and S_1 of the same size. The challenger chooses a bit $b \xleftarrow{\$} \{0,1\}$, a secret $s \in \mathbb{Z}_p^*$, and returns a group token for S_b (constructed using the secret s). The adversary wins if it correctly guesses b.

All the previous constructions [AD16, AD17, KLEL17, AD21] argued that any adversary could predict b correctly if and only if it can extract *the randomness* (the value s in line 2 of GroupGen algorithm) used to generate the challenge group token. They also argued that predicting b by any adversary is equivalent to computing the group token Γ_{S_b}, which requires the knowledge of the secret used in token generation. Moreover, predicting the secret is equivalent to computing the *discrete log* in $\mathbb{G}$. As *the discrete logarithm problem* in $\mathbb{G}$ is "hard", thus the adversarially cannot win the group privacy security with significant probability.

This security argument is incorrect. The security argument of [AD16] (similarly in [AD17, KLEL17, AD21]) says that if an adversary can break the discrete logarithm problems, then it can also break the BrED scheme of [AD16]. In provable security, the argument should be the converse, *i.e.*, if an adversary can break the *group privacy* security of [AD16] then it can also break the discrete logarithm problem in the respective group.

Looking ahead, it is, in fact, easy to check from our attack that [AD16] (similarly [AD17, KLEL17, AD21]) are not secure wrt *group privacy.* The attack idea is simple. The public parameters and the group token form a DDH instance for any adversarially chosen $S \subseteq [n]$.